COPY

AD-A220 710

190500-1-F

Final Report

SRIM USER'S MANUAL

Release 2.0

C.L. ARNOLD, Jr.
Radar Division

MARCH 1986

Naval Ocean Systems Center
271 Catalina Blvd.
San Diego, California 92152-5144

Contract No. N66001-85-C-0128

ENVIRONMENTAL
RESEARCH INSTITUTE OF MICHIGAN
BOX 8618 • ANN ARBOR • MICHIGAN 48107

SECURITY CLASSIFICATION OF THIS PAGE (*When Data Entered*)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (*and Subtitle*) SRIM User's Manual Release 2.0		5. TYPE OF REPORT & PERIOD COVERED Final Report 6. PERFORMING ORG. REPORT NUMBER 190500-1-F
7. AUTHOR(*s*) C. L. Arnold, Jr.		8. CONTRACT OR GRANT NUMBER(*s*) N66001-85-C-0128
9. PERFORMING ORGANIZATION NAME AND ADDRESS Environmental Research Institute of Michigan Radar Division P.O. Box 8618, Ann Arbor, Michigan 48107-8618		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Ocean Systems Center 271 Catalina Blvd. San Diego, California 92152		12. REPORT DATE March 1986 13. NUMBER OF PAGES v + 142
14. MONITORING AGENCY NAME AND ADDRESS (*if different from Controlling Office*)		15. SECURITY CLASS. (*of this report*) Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A

16. DISTRIBUTION STATEMENT (*of this Report*)

17. DISTRIBUTION STATEMENT (*of the abstract entered in Block 20, if different from Report*)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (*Continue on reverse side if necessary and identify by block number*)

Synthetic aperture radar
Signature simulation
Geometric modeling

20. ABSTRACT (*Continue on reverse side if necessary and identify by block number*)

This is a user's manual for producing simulated radar images (SRIM) of computer-modeled objects. It uses a solid modeler to build a target and estimates the electromagnetic scattering of the radar wave using ray-tracing and physical optics. See p 4

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

SECURITY CLASSIFICATION OF THIS PAGE (*When Data Entered*)

TABLE OF CONTENTS

APPENDICES

1
INTRODUCTION

1.1 SCOPE

The Simulated Radar IMage (SRIM) software is a prototype system consisting of several programs set up to run as a production system. This document describes how the SRIM programs are set up and run. The intent is to provide the user with the information to produce simulated radar images from a target description such as engineering drawings.

1.2 RELATED DOCUMENTS

T. B. D.

1.3 OPERATING ENVIRONMENT

All of the SRIM software is written in VAX FORTRAN 77. The DEC extensions to FORTRAN have been avoided to aid transportability. SRIM is currently running on a VAX/780 under VMS 4.x and uses a 300 Mb disk. A magnetic tape can be used instead of the disk but execution speed will be reduced.

1.4 DOCUMENT OVERVIEW

Section 2 contains an introduction to the SRIM system. It gives a qualitative description of the system and of the functions performed by the individual programs. The section includes a brief and qualitative discussion of the theory behind the system. More details on the theory are available in the references given in Sect. 1.2.

Section 3 describes the individual programs more fully. The inputs and outputs for each program are given, especially the user inputs.

Section 4 contains information needed to quantitatively define the SRIM system user inputs. The emphasis is on how the values for the user inputs are determined. The section contains an example showing how the user inputs are determined for a specific SRIM simulation.

The appendices contain detailed information on the SRIM files as well as example work sessions for all of the SRIM programs.

2
INTRODUCTION TO THE SRIM SYSTEM

2.1 SYSTEM DESCRIPTION

The interdependent parts of the SRIM system are shown in Figure 2-1. Two parts of the SRIM system are not shown. The programs can be run separately, the GIFT program must be run first since its ray history output is used by the other programs. The only other restriction on the order of execution is that the RADSIM program must be run before the DETECT program.

The GIFT program takes a target description from the geometry description file. It then uses combinatorial geometry to construct the solid model of the target from the basic solids and combinations specified in the geometry definition file. GIFT then obtains ray tracing information from the user. This information specifies the initial orientation and the ray density for the ray trace. GIFT impliments two user selected functions based on the ray trace: Generating a ray history file for use by RADSIM, or generating a SHADED optical image of the target. If a ray history is requested, The rays are traced into the target and through their reflections from the target surfaces. A ray will not be traced through more than a user-determined maximum number of reflections. The ray trace information is written to a ray history file which provides the geometry input for the other programs. If an optical image is requested, the ray trace is done only to the first reflection. The output is a displayable optical image. A ray history file is not output from this function.

The GIFTDUMP program reads the GIFT ray history file and outputs a dump of its contents to a file for printing. The user can specify that only certain ray history records are to be dumped. This reduces the output (ray history records are large). The user can also obtain a histogram of the number of rays having 0,1,2,... reflections. This program is useful for very detailed checking of ray histories.

The SHADE program produces SHADED optical images of the target. It uses the ray history file and an illumination direction specified by the user. The result is a photograph-like image of the target as seen from the initial direction specified for the rays in GIFT. This is achieved by projecting the optical image onto a plane perpendicular to the initial ray direction. These SHADED optical images are very useful for visualizing the target.

The OVERLAY program produces a SHADED image projected into the radar slant plane with a user specified illumination direction. The slant plane is perpendicular to the optical plane used by SHADE and the result is a SHADED image in range and cross-range. These are the same coordinates used in a SAR image and the OVERLAY image is

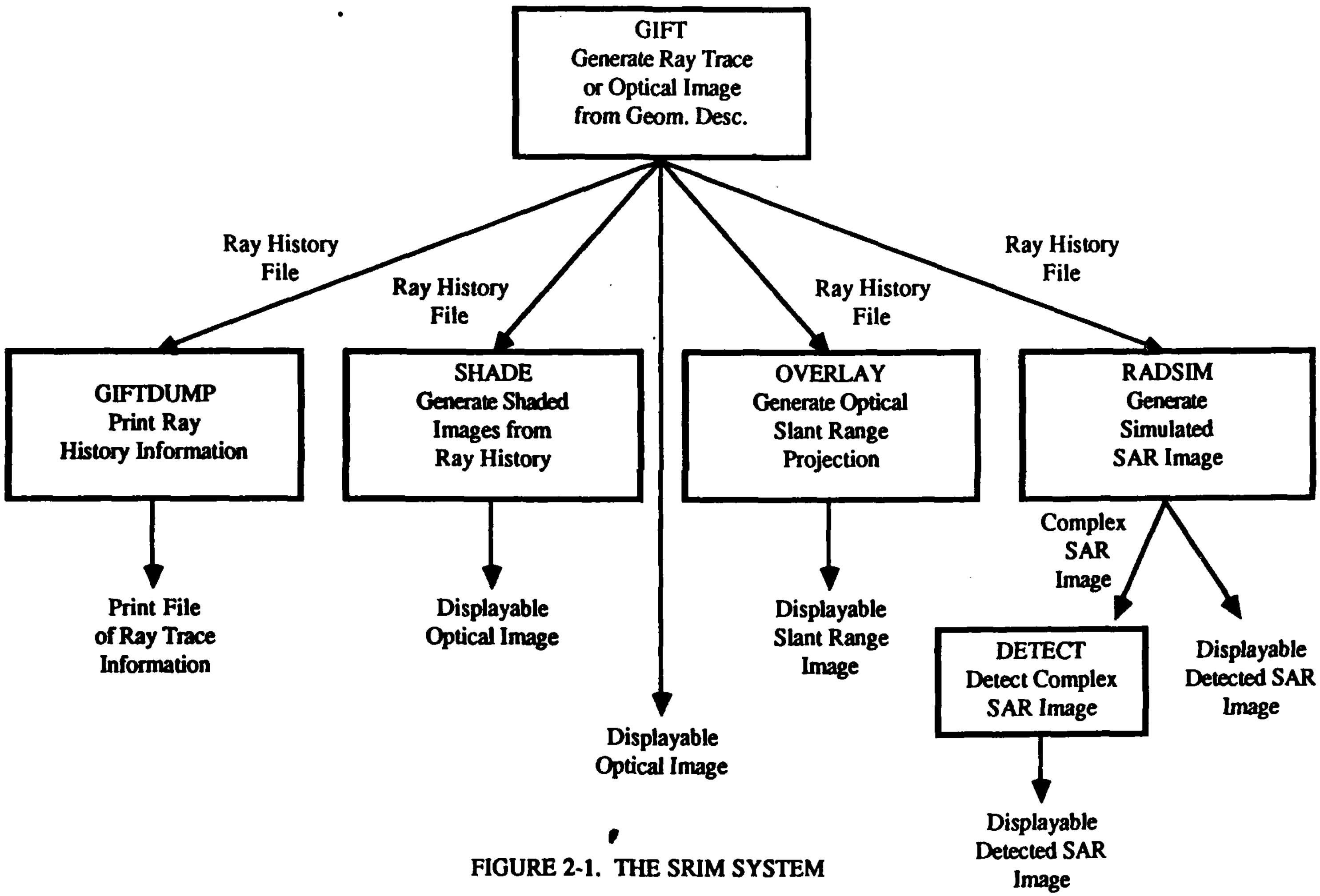


FIGURE 2-1. THE SRIM SYSTEM

useful for seeing the geometric form of this slant range projection.

The RADSIM program produces simulated SAR images. Its geometric information comes from the GIFT ray history file. This allows it to estimate the scattering of the radar waves by the target. This estimate of electromagnetic scattering is combined with other information about the radar to simulate the SAR image. The user defines the image by giving the size of a pixel (in distance units) and the image size (in pixels). The user also supplies the following input files:

-> A file defining the radar. This file contains the radar wavelength, resolution, polarization, etc.

-> A file defining the reflection models. For example, a model to use for perfect reflectors, a modelfor absorbers, etc.

-> A file specifying which reflection model to use for the various surfaces of the target.

The output of RADSIM is a complex SAR image. Normally, a displayed image will be magnitude detected.

The DETECT program takes the RADSIM complex image and detects it. This is done by multiplying the value for each pixel by a scale factor and then finding its amplitude. The scale factor can be given by the user or found automatically by DETECT. The output is a detected display file.

2.2 THE GIFT PROGRAM

Figure 2-2 shows the inputs and outputs for GIFT. The user can request two functions from GIFT: Produce a ray history file, or produce a SHADED optical image of the target. If the SHADED optical image is requested, GIFT does not create a ray history file. The inputs that will be discussed are the geometry description file and the part of the interactive input that defines the initial direction and spacing of the rays. The geometry description file tells GIFT how to construct a solid model of the target. The geometry description is in terms of combinatorial geometry. This is a method of solid modeling that builds a solid by combining elemental solids using Boolean rules. The geometry definition file contains a list of the elemental solids (called primitives) that make up the target and the rules for combining these solids to build the solid regions that make up the target. Some of the primitives are shown in Figure 2-3. The primitives are combined using the Boolean operations: intersection, union, and difference. These operations are illustrated in Figure 2-4. A complete list of the primitives available to GIFT is shown in Appendix A. The regions are the solid model of the target. In other words, the target consists of an implicit union of regions which are constructed by Boolean operations applied to primitives. Figure 2-5 shows an example of a simple target consisting of one region which is built from three primitives.

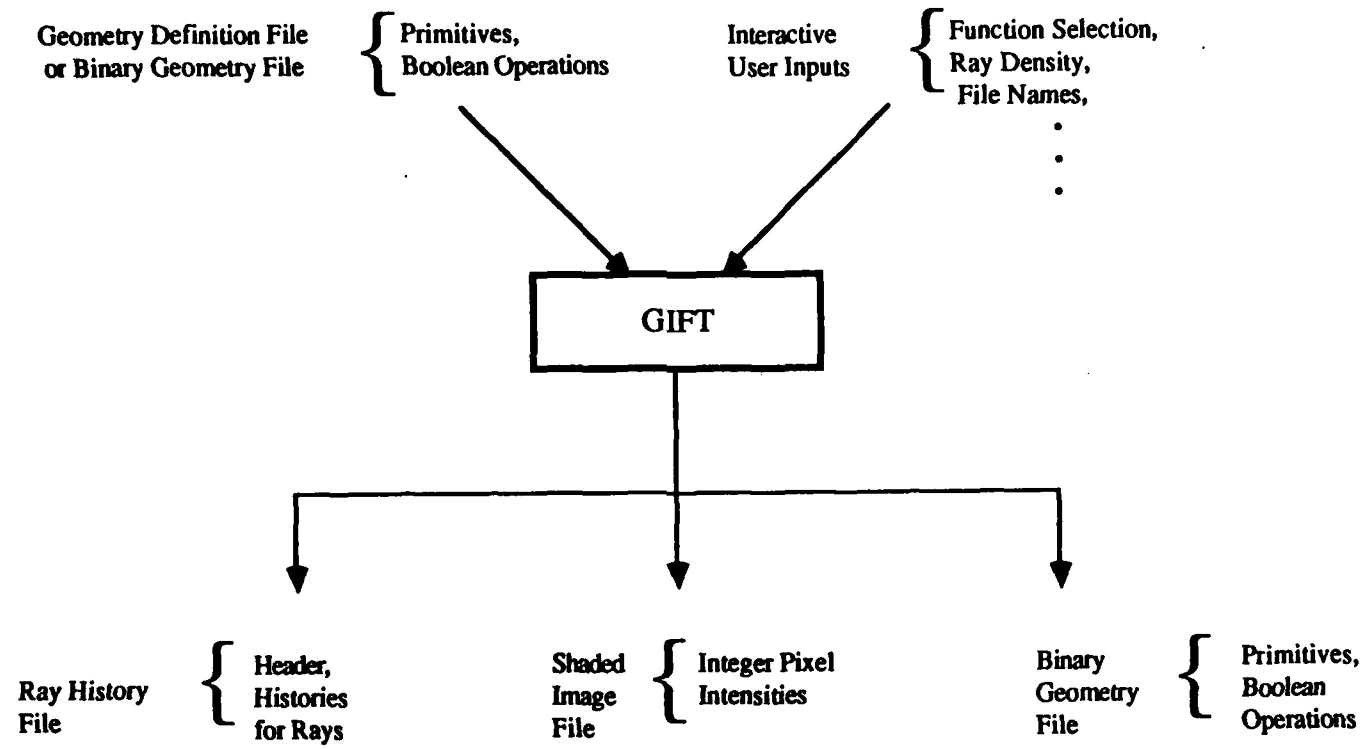


FIGURE 2-2. GIFT INPUTS AND OUTPUTS

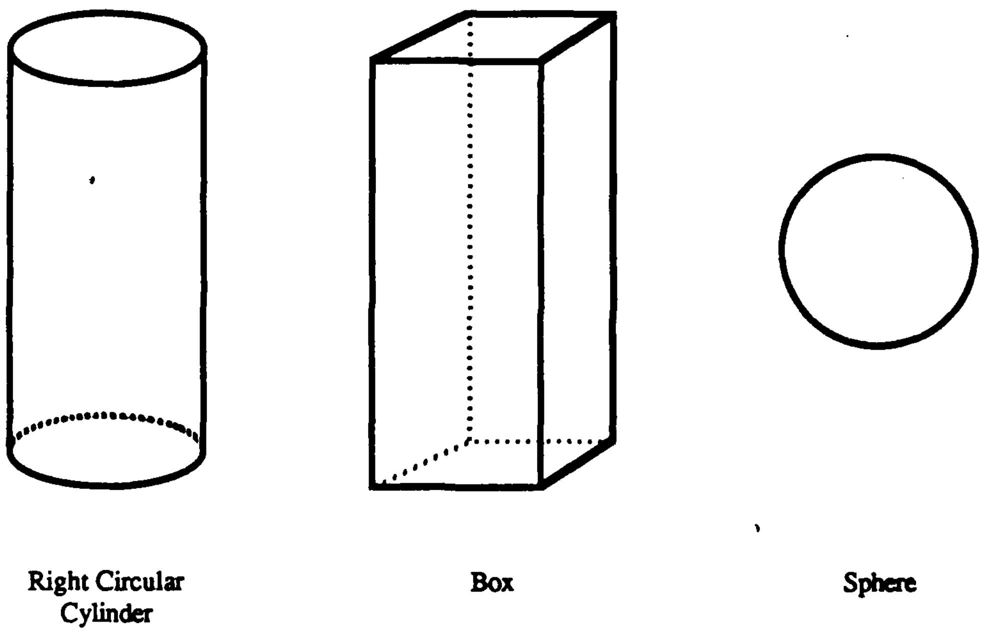


FIGURE 2-3. SOME SOLID GEOMETRY PRIMITIVES

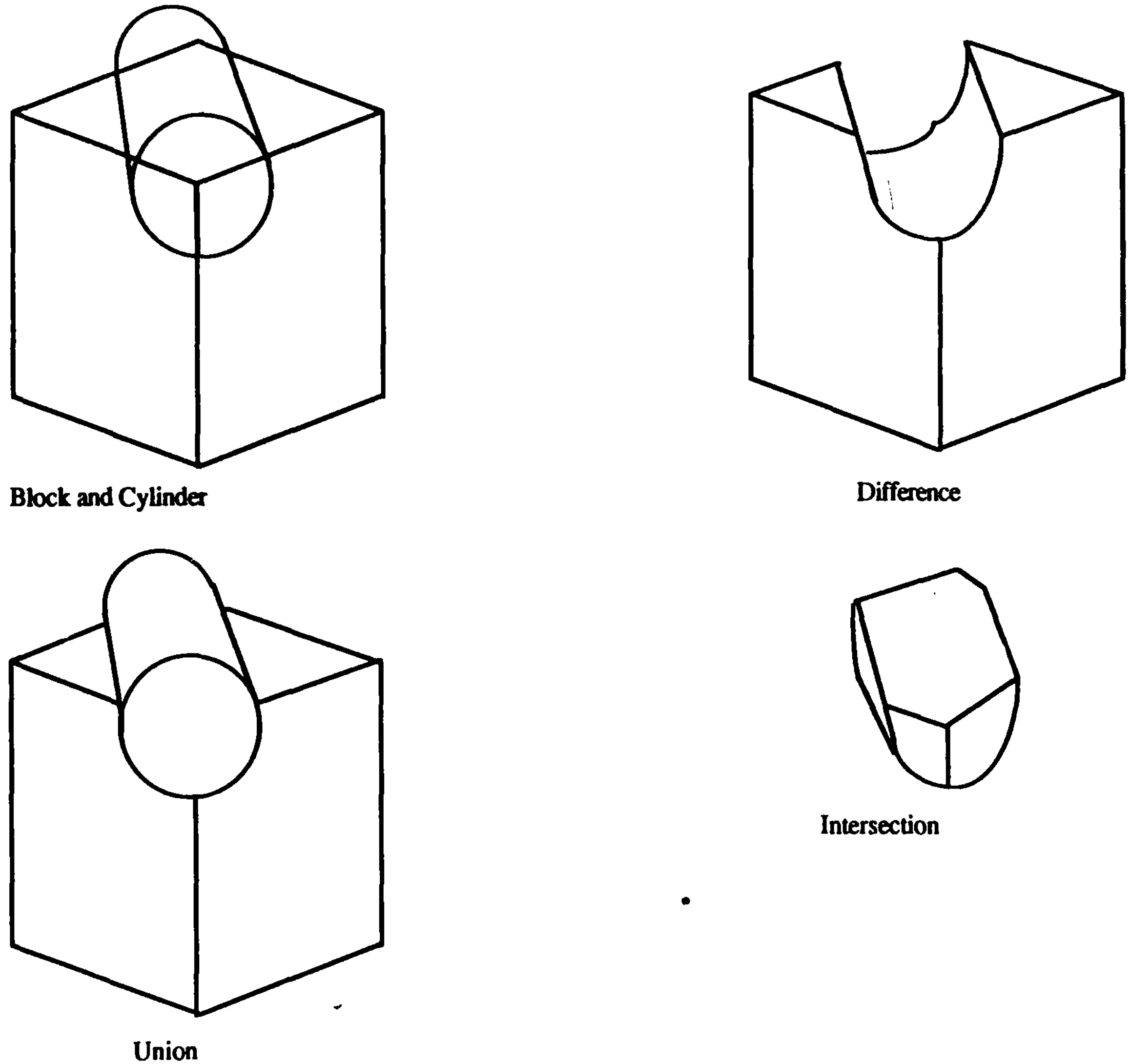


FIGURE 2-4. BOOLEAN OPERATIONS

After GIFT has read the geometry definition file into its internal data area, it writes out a binary geometry file. This file is a binary file containing GIFT's internal geometry structure. If the same geometry is run again, the binary geometry file can be given as the input to GIFT and this will speed up the run since GIFT can read in the binary geometry file much faster than it can read in the geometry definition file.

After GIFT has built the solid model of the target, the user must tell it the initial direction and density of the rays to be traced. It is assumed that the radar is far from the target and that the incident wave is planar. Thus any properly oriented plane represents a wave front and can be used as a reference plane for the scattering problem. This plane is called the emanation plane. GIFT will start (fire) its rays from this plane, directed along the plane normal. The normal to the emanation plane is specified by the user. The user gives the values of the two angles shown in Figure 2-6. These angles are the usual spherical coordinate angles. GIFT determines the range coordinate and sets it so that the emanation plane is at the minimum range such that the plane does not intersect the target. The emanation plane contains an emanation rectangle. This rectangle is positioned and sized so that a projection of the target onto the emanation plane is just bounded by the rectangle. The cartesian coordinate system shown in Figure 2-6 is the overall coordinate system for SRIM and is the coordinate system used to set up the target geometry.

Once the emanation rectangle is established, GIFT will inform the user of the emanation rectangle size and prompt for the number of rays in the vertical and horizontal directions. The horizontal direction is along an edge of the emanation rectangle that is parallel to the x-y plane. The vertical direction is along an edge that is perpendicular to the horizontal direction and lies in the emanation plane (note that this direction is not, in general, perpendicular to the x-y plane). These directions establish an emanation plane coordinate system as shown in Figure 2-6. In Figure 2-7, an emanation rectangle is shown with the starting points for a 3 by 4 grid of rays. An emanation plane coordinate system is shown in Figure 2-7 to give the directions of the coordinate system axis. However, the emanation plane coordinate system will not generally be centered in the emanation plane rectangle. The rays are numbered in the order in which they are generated. The starting points of the rays are called firing points, and rays are fired from these points parallel to the emanation plane normal. Along with the horizontal and vertical emanation rectangle coordinates, GIFT assigns row and column numbers to the firing points. In Figure 2-7, there are 4 columns and 3 rows. The firing order in terms of (col,row) is (1,1), (2,1), (3,1), (4,1), (1,2), ... which gives a location for the 8th ray of (4,2).

If the ray history function has been requested, The coordinates of the firing point and other initial

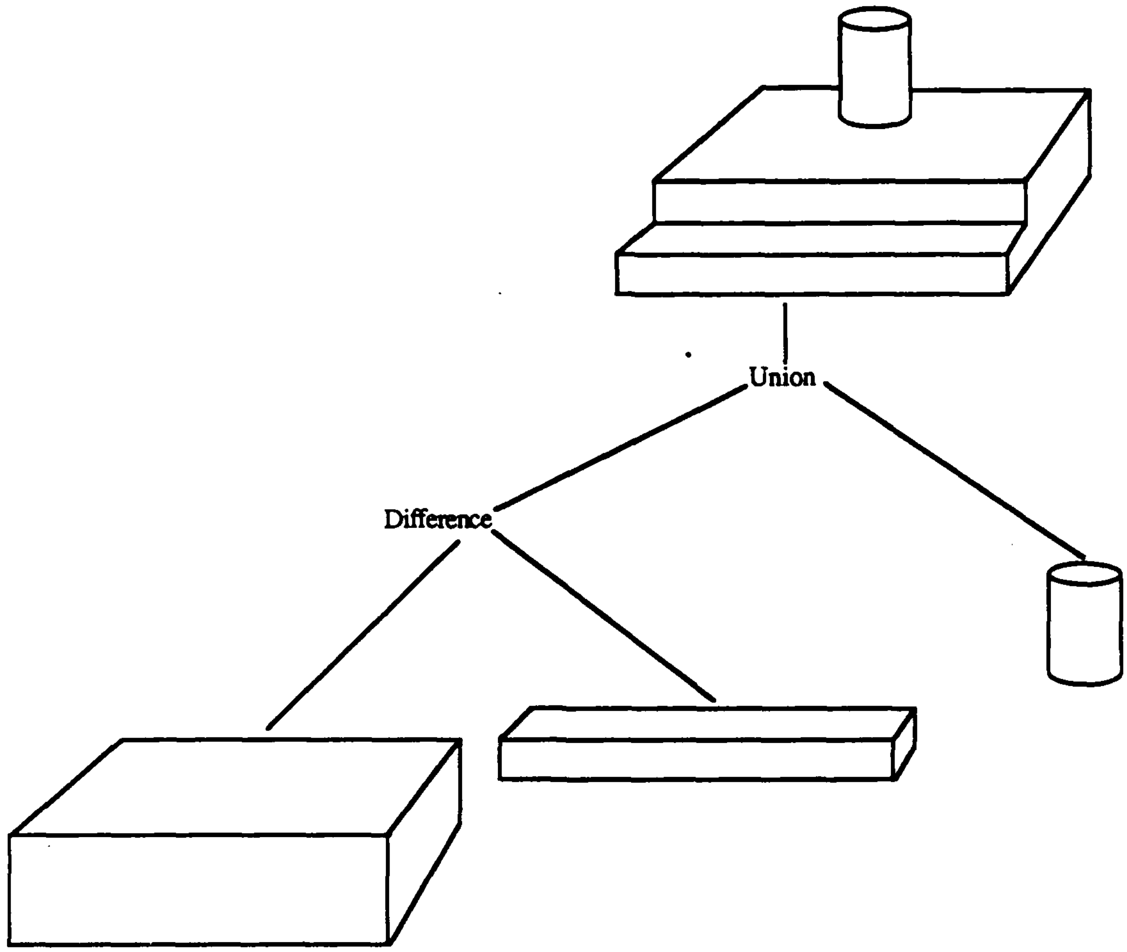
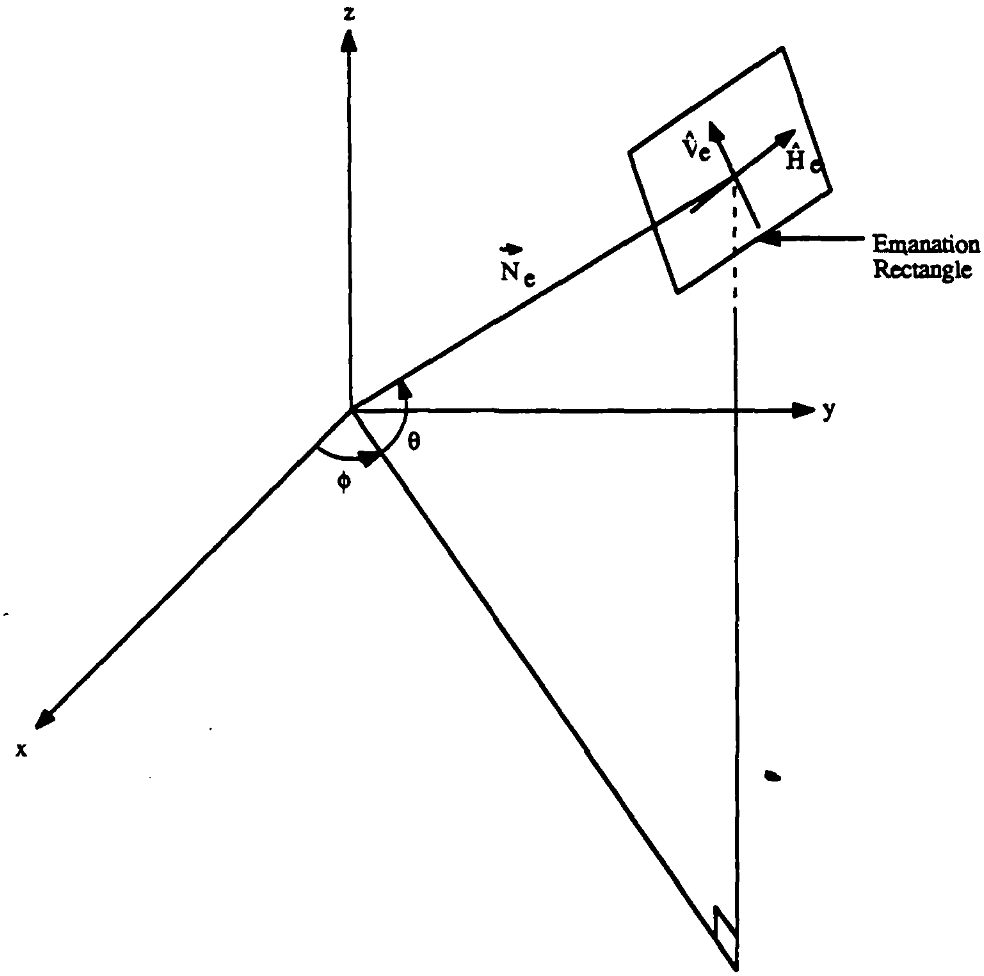


FIGURE 2-5. BUILDING A SIMPLE OBJECT USING ONE REGION AND THREE PRIMITIVES.



φ = Azimuth

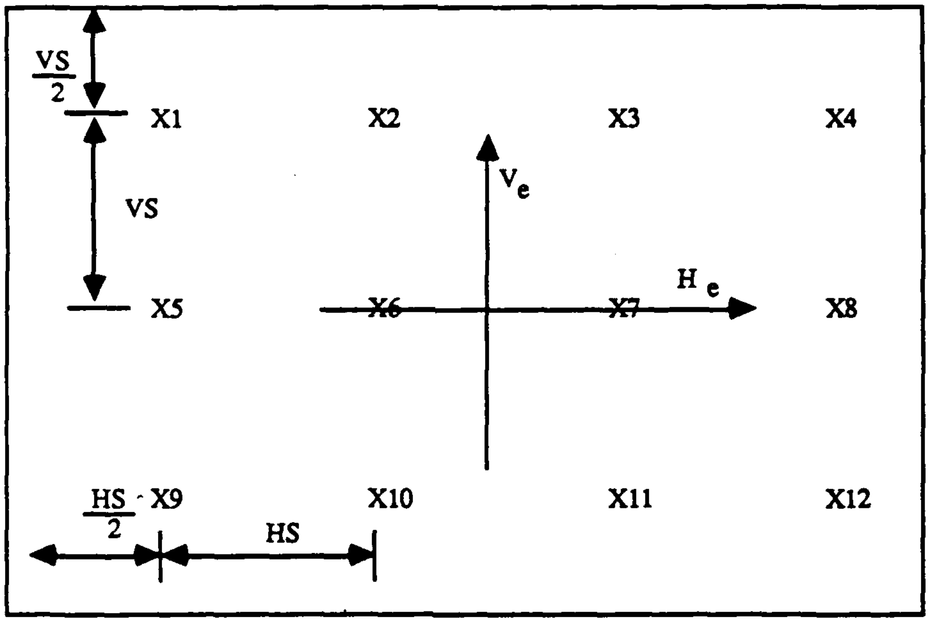
θ = Elevation

$\vec{N}_e$ = Emanation Plane Normal

$\hat{H}_e$ = Horizontal-Axis Unit Vector for Emanation Plane Coordinate System

$\hat{V}_e$ = Vertical-Axis Unit Vector for Emanation Plane Coordinate System

FIGURE 2-6. EMANATION PLANE DEFINITION; NOTE THAT THE EMANATION RECTANGLE IS NOT NECESSARILY CENTERED ON $\vec{N}_e$



H_e = Horizontal Axis

V_e = Vertical Axis

HS = Horizontal Ray Spacing

VS = Vertical Ray Spacing

X1...X12 = Firing Points for the Rays

FIGURE 2-7. EMANATION RECTANGLE

information is recorded in a ray firing record in the ray history. After a ray is fired, it is traced through its reflections from the target surfaces. At each reflection, the reflection point coordinates, surface identifier, surface normal, and other information is found. This information is written to the ray history as a reflection record. When the ray exits the target without any further reflections, an escape record is written to the ray history file.

The ray history contains header information consisting of run parameters (such as emanation rectangle size and number of rays fired), followed by ray trace histories for each ray. The ray trace histories contain a firing record, a reflection record for each reflection, and usually an escape record. The contents of the ray history file are detailed in Appendix B.

If the shaded image function is requested, GIFT will produce a shaded optical image of the target as viewed from the emanation rectangle. Each ray corresponds to a pixel in the image. Thus a 128X128 grid of rays fired by GIFT will result in a shaded optical image of 128X128 pixels. The user inputs are the file names for the input ray history and output image files as well as the illumination direction (for shading). The illumination direction is specified using the angles shown in Figure 2-6 (azimuth and elevation). This determines the unit vector from the center of the coordinate system to the light source. The pixel word size is determined by the number of intensity levels that can be displayed by the display system. The pixel word size is set to 8 bits so that each pixel will contain an integer in the range 0 to 255. The ray trace is only carried out to the first reflection. The dot product of the surface normal and the illumination direction is used to establish an intensity for the pixel represented by the ray. The result of the dot product is then converted to an integer scaled to the pixel word size.

2.3 GIFTDUMP

The inputs and outputs for the GIFTDUMP program are shown in Figure 2-8 This program writes ray history information to a file for printing. The information to be output can be selected by the user to be:

1. Only specific physical records.
2. Only rays whose number of reflections is equal to or greater than a user given number.
3. A histogram of the number of rays with 0, 1, 2, 3, ... reflections.

For example, the user can dump all rays with 2 or more reflections from physical record number 12. A physical record is very large and contains the history for a number of rays. Therefore, if the requested information is not limited by the user options, the dump can become impractically large. The dump will contain the ray history

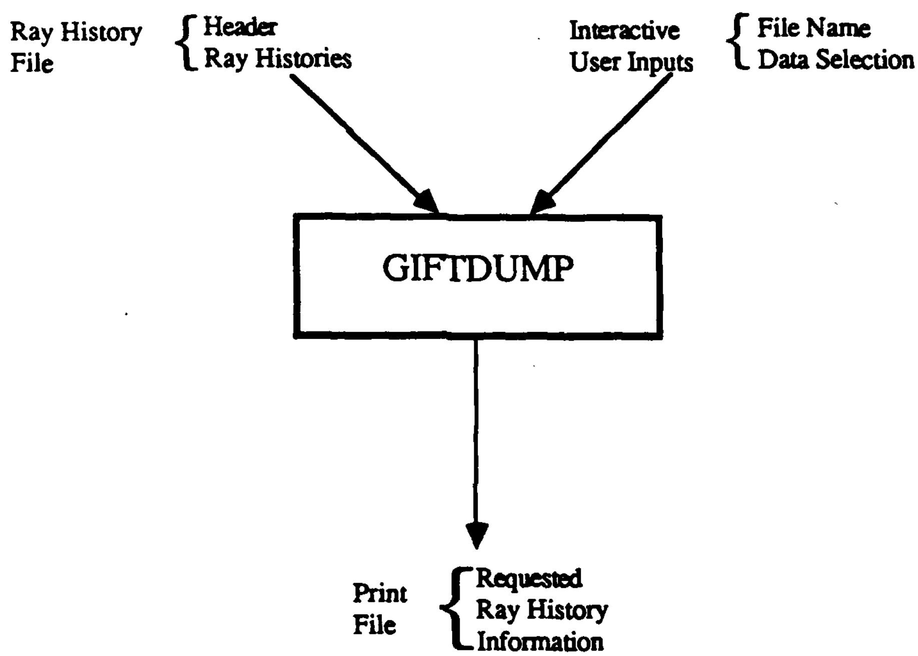


FIGURE 2-8. GIFTDUMP INPUTS AND OUTPUTS.

file header information followed by the information requested by the user.

2.4 SHADE

The inputs and outputs for SHADE are shown in Figure 2-9. SHADE takes the GIFT ray history file as its primary input and outputs a SHADED optical image of the target as viewed from the emanation rectangle. Each ray corresponds to a pixel in the image. Thus a 128X128 grid of rays fired by GIFT will result in a SHADED optical image of 128X128 pixels. The user inputs to SHADE are the file names for the input ray history and output image files as well as the illumination direction (for shading). The illumination direction is specified using the angles shown in Figure 2-6 (azimuth and elevation). This determines the unit vector from the center of the coordinate system to the light source. The pixel word size is determined by the number of intensity levels that can be displayed by the display system. The word size is set to 8 bits and each pixel will contain an integer in the range 0 to 255.

SHADE processes the ray history in the following steps:

I. Obtain the file names and illumination direction from the user.

II. For each ray in the ray history.
   - A. If the ray does not reflect off of a target surface, go to the next ray.
   - B. If the ray has reflections, then for the first reflection
     1. Pick up the surface normal from the reflection record.
     2. Find the dot product of the surface normal and the illumination direction.
     3. Convert the result of step 2 to an integer in the range 0 to 255.
     4. Place the integer in the image location corresponding to the row and column of the ray firing position.

III. Write the image to the output file.

2.5 OVERLAY

The inputs and outputs for the OVERLAY program are shown in Figure 2-10. The OVERLAY program also produces a SHADED image from a GIFT ray history. However, the image is projected into the slant range plane and is not an optical image. The slant range plane is perpendicular to the emanation plane and contains, at broadside squint, the horizontal axis of the emanation rectangle coordinate system. The image position for the contribution from a ray is computed by finding the range and azimuth of the first

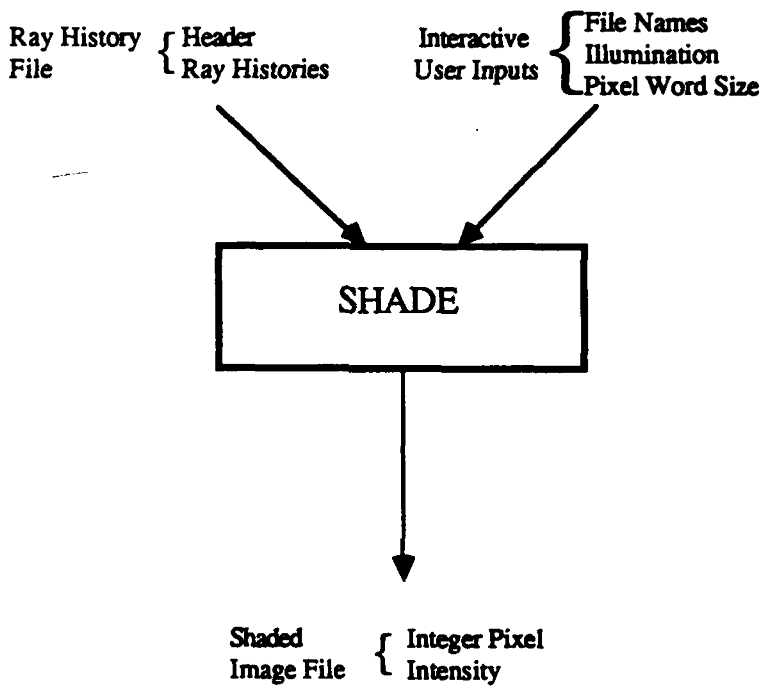


FIGURE 2-9. SHADE INPUTS AND OUTPUTS

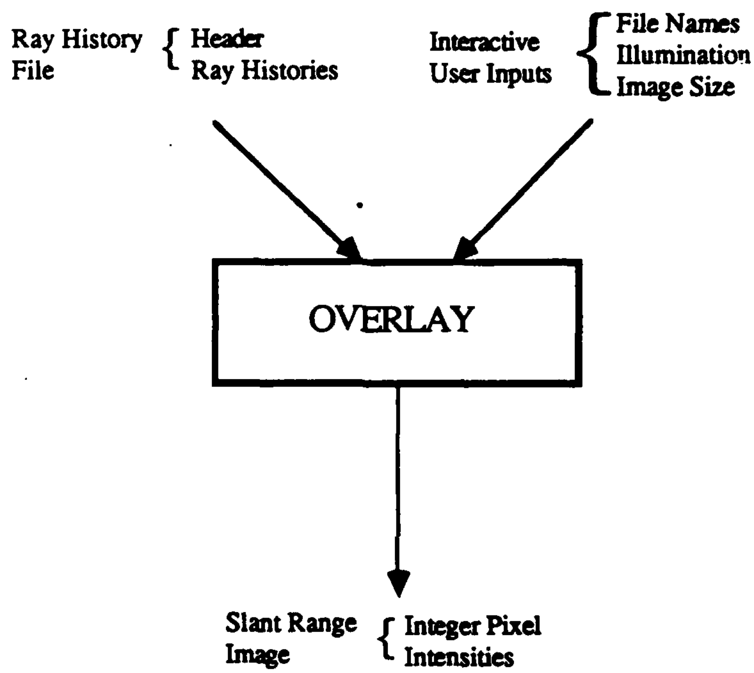


FIGURE 2-10. OVERLAY INPUTS AND OUTPUTS

reflection and converting these coordinates into a pixel location. The user input illumination direction is identical to the corresponding SHADE input. The user must also supply the image size in pixels and the pixel size. For example, if the user specifies an image size of 128X128 pixels and a pixel size of .5 feet, then the image will cover 64X64 feet in the slant range plane centered on the origin of GIFT's overall coordinate system.

OVERLAY processes the GIFT ray history in the following steps:

I. Obtain the file names, illumination direction, and the image information from the user.
II. Determine the emanation plane normal and the azimuthal direction from the ray history header information.
III. For each ray in the ray history
   A. If the ray has no reflections, go to the next ray.
   B. If the ray has reflections, then for the first reflection.
      1. Pick up the surface normal from the reflection record
      2. Find the dot product of the surface normal and the illumination direction
      3. Find the range and azimuth of the reflection and convert to a pixel location
      4. Add the dot product result to the pixel location.
IV. Convert the image pixels to integers in the range 0 to 255 and write the image to the output file

Note that, unlike SHADE, more than one ray can contribute to a pixel value.

2.6 DETECT

The inputs and outputs for the DETECT program are shown in Figure 2-12. The DETECT program converts a RADSIM complex image into a detected image. The input is a RADSIM complex image where each pixel contains a complex real number. The output is an image where each pixel contains a 16 bit integer which is the magnitude of the corresponding complex number. The user inputs are the file names for the input and output files, and a scale factor selection. The scale factor can be given by the user or can be determined by DETECT. The scale factor allows the user to raise the level of the pixel values so that weak returns are not lost when the pixel values are converted to integers. DETECT processes the image in the following steps:

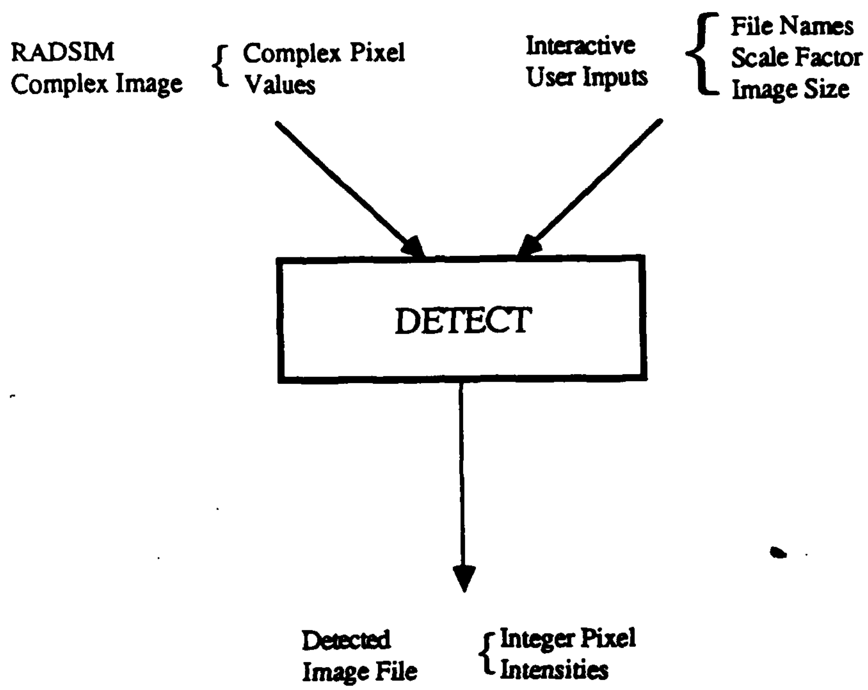


FIGURE 2-12. DETECT INPUTS AND OUTPUTS

I. Collect the user inputs.
II. Read in the complex image.
III. For each pixel in the image
   A. Compute the magnitude squared of the complex number.
   B. Multiply the result of A by the scale factor.
   C. Truncate the result of B to a 16 bit integer.
IV. Write out the detected image.

If the user gives the scale factor, it is the users responsibility to chose the scale factor so that the detected image will have intensities in the proper range for display.

2.7 THE RADSIM PROGRAM

Figure 2-13 shows the inputs and outputs for RADSIM. Section 2.7.1 will discuss the method used by RADSIM to compute the returns and create the complex image while Section 2.7.2 will cover some of the input files. The inputs discussed in Sect. 2.7.2 are the surface type file, the radar file, and the reflection model file. The user inputs will be covered in section 3.7 and the ray history file from GIFT is covered in detail in Appendix B.

2.7.1 GENERATING THE IMAGE

RADSIM uses the ray history from GIFT and geometric optics to estimate the total electromagnetic fields at the target surfaces including contributions from multiple reflections. It then uses physical optics to find the scattered field (return) at the radar receiver. This estimate of the return along with knowledge about SAR data processing and the system point response is used to determine the complex image. This image may (by user option) be written out as either a scaled real image or as a complex image. The ray history is a discrete sampling of the target's geometry and this leads to ray density criteria which are discussed in Section 4.2. The procedure for finding returns from the rays is applied to each ray in the ray history.

The overall slant plane imaging geometry assumed by RADSIM is shown in Figure 2-14. The slant plane is the plane defined by the aircraft velocity vector and the motion compensation point (usually at the center of the target). The image of the target is in the slant plane with coordinates of range and azimuth (cross-range) and is centered at the motion compensation point. The azimuth axis is parallel to the aircraft velocity vector. A target is imaged by a side looking SAR (squint angle is 90 deg.). The vector from the center of the aperture to the center of the target will be the emanation plane normal. Since the distance to the radar is much greater than the dimensions of

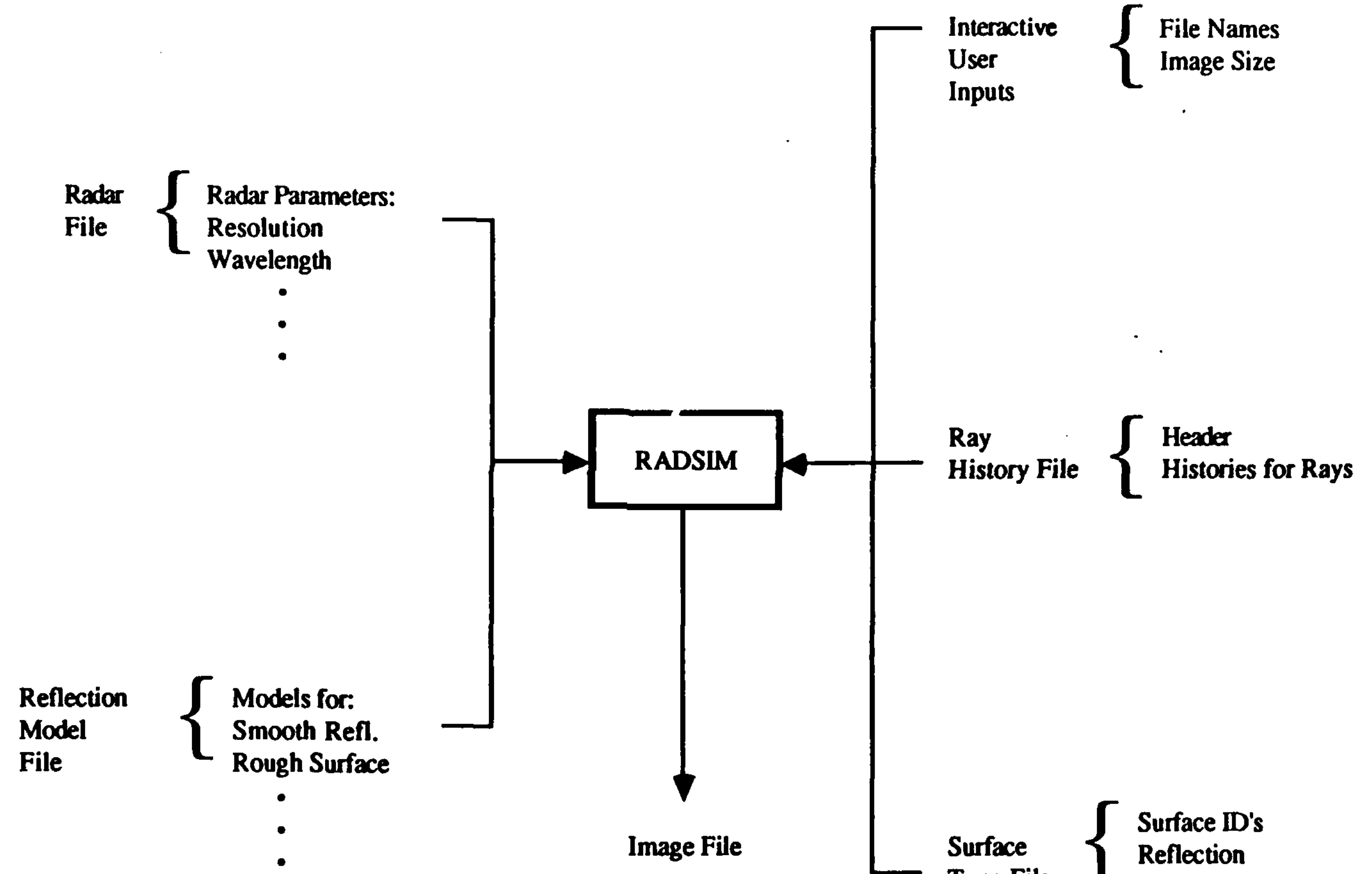


FIGURE 2-13. RADSIM INPUTS AND OUTPUTS

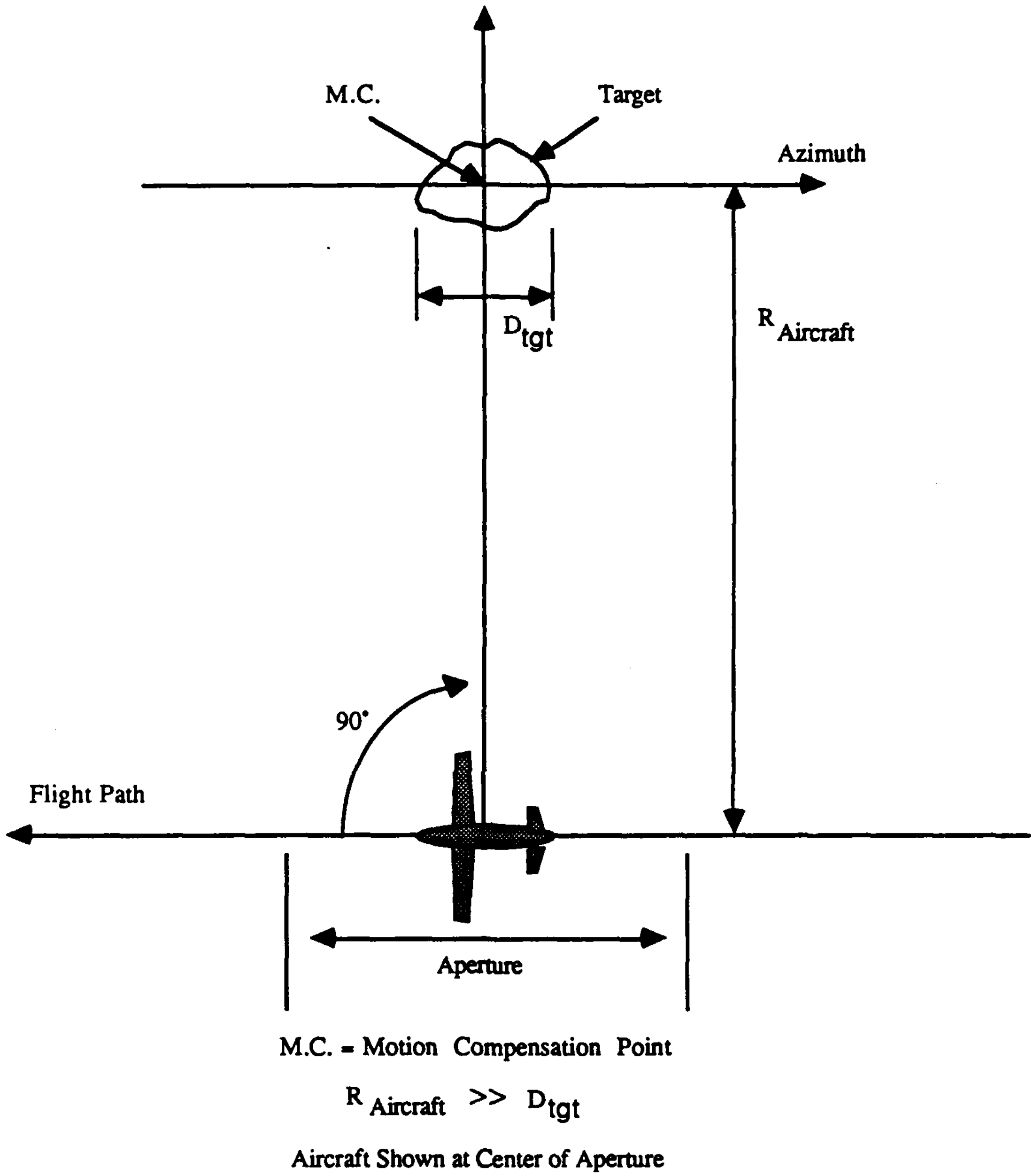


FIGURE 2-14. RADSIM IMAGING GEOMETRY IN SLANT RANGE PLANE

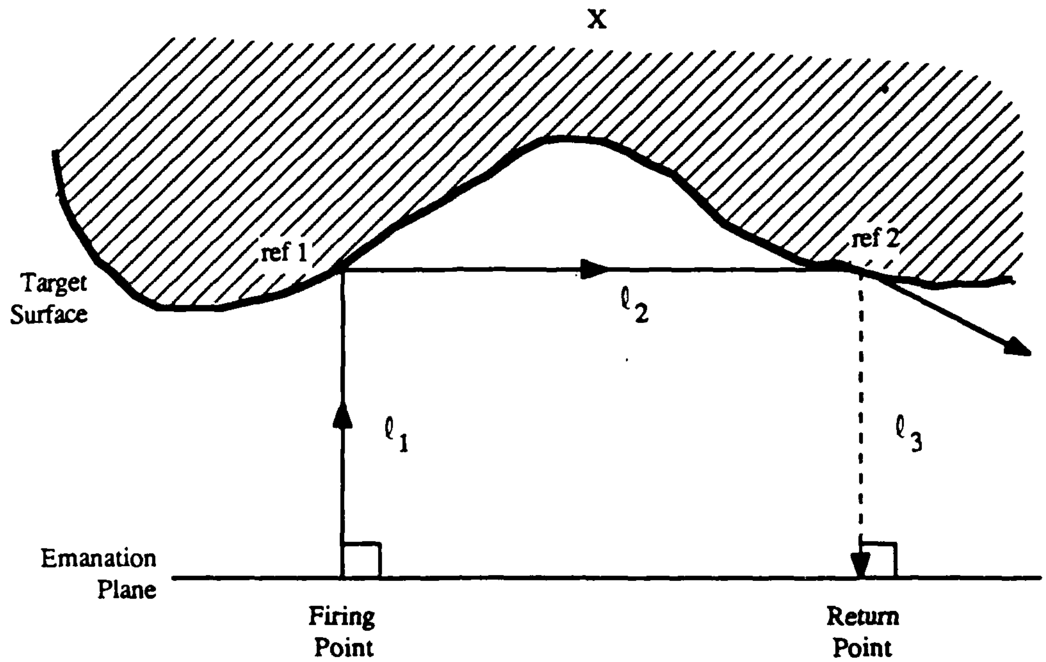
the target, the incident wavefront is approximately planar and can be represented by an array of rays fired from the emanation plane rectangle as described in Section 2.2. Given the reflection history of the rays, knowledge of SAR processing is used to determine where the return will be placed in the complex image.

The situation for a typical ray is shown in Figure 2-15. RADSIM places the return at 1/2 of the total ray path length. For a singly reflected return, this is the distance from the radar to the reflection point on the target. Because of the doppler processing done in a SAR, a return is placed in azimuth (cross-range) at a point which is the average of the first and last reflection azimuths. For a singly reflected return, this corresponds to the azimuth position of the reflection point on the target. Note that for a multiply reflected return such as shown in Figure 2-15, the return is assigned a range and azimuth that does not correspond to a physical point on the target. This is the source of many of the artifacts in SAR images. The phase of the complex return from a reflection is the phase due to the total path length for the ray (phase due to range) plus any phase shifts due to the scattering at the reflection points. This is illustrated in Figure 2-15.

Figure 2-16 shows the image points in the slant range plane. The range and azimuth axis are shown with the integer coordinates of the points which are the pixel numbers in the image. The axis are oriented as they are on the image display. RADSIM sets up the image so that the origin of the coordinates is in the lower left corner of the image and the origin of the coordinate system shown in Figure 2-6 is in the center of the image. Contributions to the image are assigned to an image point and all contributions assigned to the same image point are coherently summed to determine the complex image value for the corresponding image pixel. The azimuth direction is the same as the emanation rectangle horizontal axis except that the direction is reversed. This reversal is done to simplify comparisons with the optical images produced by SHADE.

As a consequence of using the physical optics approach, an accurate result for bodies with curved surfaces such as spheres and cylinders requires a very high ray sampling density. This is required to obtain the correct phase cancellation for returns away from the specular point (the point where an incident ray is specularly reflected to the radar). In order to reduce the required ray spacing, a stationary phase approach is used. A small area about the specular point is defined such that all rays reflected from within this area will have close to the same range. This is refered to as the stationary phase region and is a disk for a sphere and a narrow rectangle for a cylinder. RADSIM only counts returns from reflections within this region. Two examples of stationary phase regions for single reflection rays are shown in Figure 2-17.

The system point response is included in the image by



Solid Line is Ray Path from GIFT
Dotted Line is Path of Physical Optics Return
For the Return from the Second Reflection:

Range $= (\ell_1 + \ell_2 + \ell_3)/2$

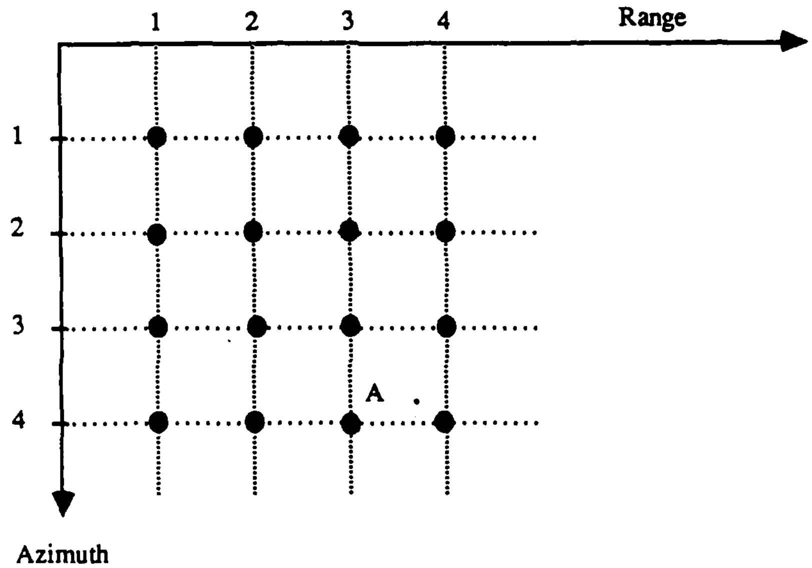
Azimuth = (Firing Azimuth + Return Azimuth)/2

Phase $= 2\pi(\ell_1 + \ell_2 + \ell_3)/\text{Wavelength} + \theta_1 + \theta_2$

Where θ_i = Phase Shift at the ith Reflection

X = Apparent Location of Return

FIGURE 2-15. SLANT RANGE VIEW OF A RAY AND THE RETURN FROM ITS SECOND REFLECTION.



The dots represent image points. The pixels for these points have image positions given by the integer values of the coordinates. For example, image point A corresponds to pixel (3,4) in the image.

FIGURE 2-16. IMAGE POINTS AND PIXEL NUMBERS

performing input centered convolutions with the system point response function. Since phase cancelling between rays is very sensitive to range differences, the range convolution is done on each ray after it is assigned an azimuth pixel number and the convolution distributes the return among the relevant range pixels. The output points of the range convolution are chosen to correspond to range pixel coordinates. This is illustrated for a return in Figure 2-18. After all of the rays have been processed, the azimuth convolution is done. Note that there is no phase variation associated with the azimuthal direction so assigning the returns to azimuthal pixel before convolution does not introduce phase errors.

The image is built up in the following steps:

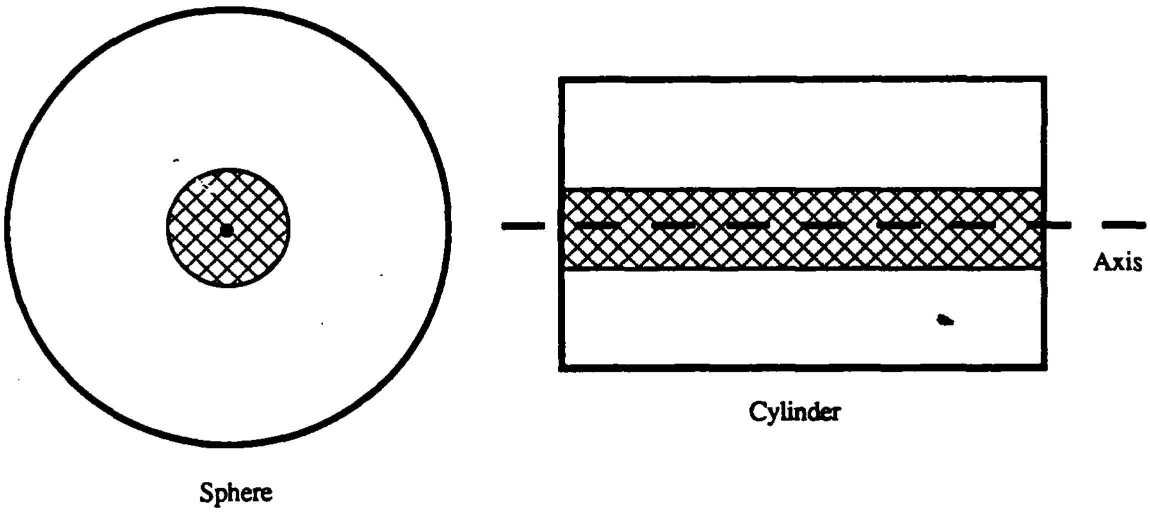
I. Loop over the rays in the GIFT ray history file.

II. For each ray:
   - A. Initialize the ray amplitude,phase, and polarization to the values given for the transmitter and initialize the ray direction to the emanation plane normal.
   - B. Move along the ray and at each reflection:
     1. Compute the amplitude, phase shift, and polarization of the specularly reflected ray using GO.
     2. If the surface is flat or the ray is within a stationary phase region, then use the incident GO field and target surface geometry at the reflection point to find the amplitude,phase shift and polarization of the physical optics field in the receiver direction.
     3. Use the total ray path length to find the phase due to range for the return and add the reflection point phase shifts to find the total phase of the return at the emanation plane.
     4. Apply the receiver phase and polarization direction to find the complex return from this reflection.
     5. Compute the image range and azimuth coordinates for the return.
     6. Convolve the return with the system range response and sum the contributions into the complex image.

III. After all rays are processed, convolve the complex image with the system azimuth response.

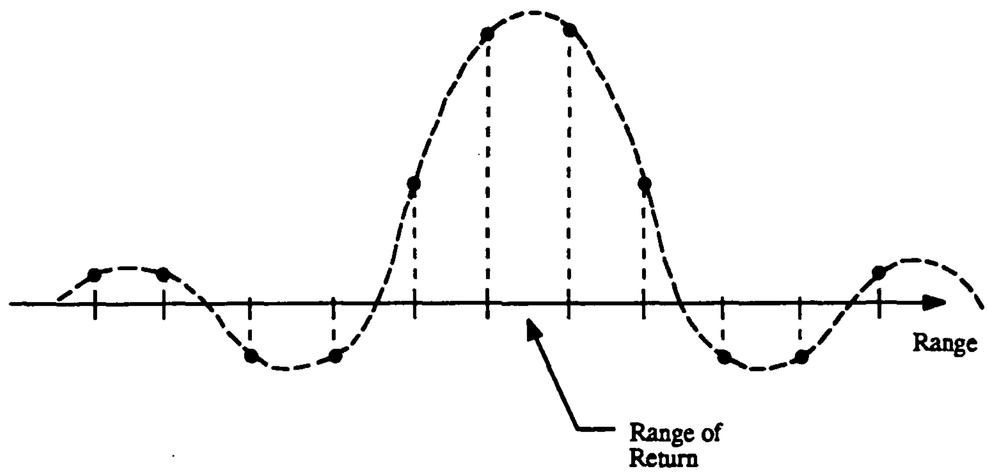
IV. If the user requested a scaled real image, convert the complex image to magnitude image scaled to be in the integer range 0 to 255 (pixel word size of 8 bits).

V. Write out the scaled real or complex image.



Cross-hatched Areas are the Stationary Phase Regions

FIGURE 2-17. STATIONARY PHASE REGIONS



The tick marks are image points and the dots are response sample points. The product of the complex return and the system point response is sampled at the image points.

FIGURE 2-18. INPUT CENTERED RANGE CONVOLUTION FOR A RETURN

2.7.2 THE RADSIM INPUT FILES

In addition to the ray history, RADSIM requires three input files. These files are the surface type file, the radar file, and the reflection model files. These files are described in detail in Appendix G.

The surface type file associates a reflection model with individual surfaces of the target. When GIFT builds the solid model of the target, each surface is assigned a unique ID. In the ray history file, each reflection record contains the ID for the reflecting surface. The surface type file associates a reflection model with the surfaces so that RADSIM can use the correct reflection model (smooth reflector, rough surface, etc.). If a surface is not assigned a reflection model in the surface type file, it defaults to a smooth perfect conductor.

The radar file defines the radar characteristics for RADSIM. It specifies the radar resolution, wavelength, transmit and receive polarization directions, system noise, and system point response. The transmitter and receiver polarization directions can have any orientation in the emanation plane and do not need to be the same. However, only linear polarization is allowed. The polarization directions are specified by giving the polarization vectors in the emanation plane coordinates (see Figure 2-7). The radar point response is assumed to be separable (same in range and azimuth) and can be unweighted or taylor weighted.

The reflection model file defines the reflection models that are used by RADSIM. These models are associated with specific surfaces of the target by the surface type file. There are 5 models that are currently defined:

1. Smooth reflector
2. A 50 percent absorber.
3. Perfect absorber
4. Statistical Gaussian ground clutter.
5. Fractal surface (for ground clutter).

The first model corresponds to a smooth perfect conductor and is a good model for smooth metal surfaces. The second model makes the surface "absorb" 50 percent of the amplitude of the incident ray. The third model makes the surface a "black hole" for rays (no returns or reflections allowed). The fourth model is a ground clutter model based on a Gaussian distribution for the returns. The fifth is a ground clutter model based on fractals. These models are discussed in more detail in Appendix H.

3
RUNNING THE SRIM PROGRAMS

3.1 LOCATIONS OF THE SRIM PROGRAMS

The VMS file structure for the SRIM system is shown in Figure 3-1. This is a tree structure where the boxes represent disk account or sub-account names and the leafs are SRIM programs. The path to a program is constructed by following the tree to the program name. For example, RADSIM is in the file [SRIM.RADSIM]RADSIM.EXE. In a VMS run command, the .EXE is not required.

3.2 GIFT

Figure 2-2 shows the inputs and outputs for GIFT. The running of GIFT will be illustrated with runs done on a simple target consisting of a sphere of radius 1 centered at (0,1,0) in the overall GIFT coordinate system. When user inputs appear in the Figures, these inputs will be underlined. GIFT expects the user inputs (except for file names) to be in lower case. The file names can be upper or lower case.

The run command to execute GIFT is

RUN [SRIM.GIFT]GIFT

As soon as GIFT starts, it will prompt the user with

Enter geometry file name: ?

The file name given to GIFT will be either the geometry definition file or the binary geometry file. Once this is done, GIFT prints its startup message and lists a set of options.

The GIFT startup message and options are shown in Figure 3-2. Most of the options are debug print options and output information on GIFT's internal data structures. These are not useful for production running of GIFT. However, two of these options will give useful information on the correctness of the geometry file setup. These options are:

s-› This will output the primitives from the geometry description file. It can be used to verify that the primitives were properly defined in the input file. It also shows the internal storage used by the primitives.

r-› This will output information on the regions. GIFT determines the minimum rectangular parallel piped (RPP) that encloses each region. This then gives the minimum and maximum values for the region along the overall coordinate system axes which are output to the user. The RPP for the entire target is also output as well as the storage space used by the regions.

Figures 3-3 and 3-4 show the outputs from these options for

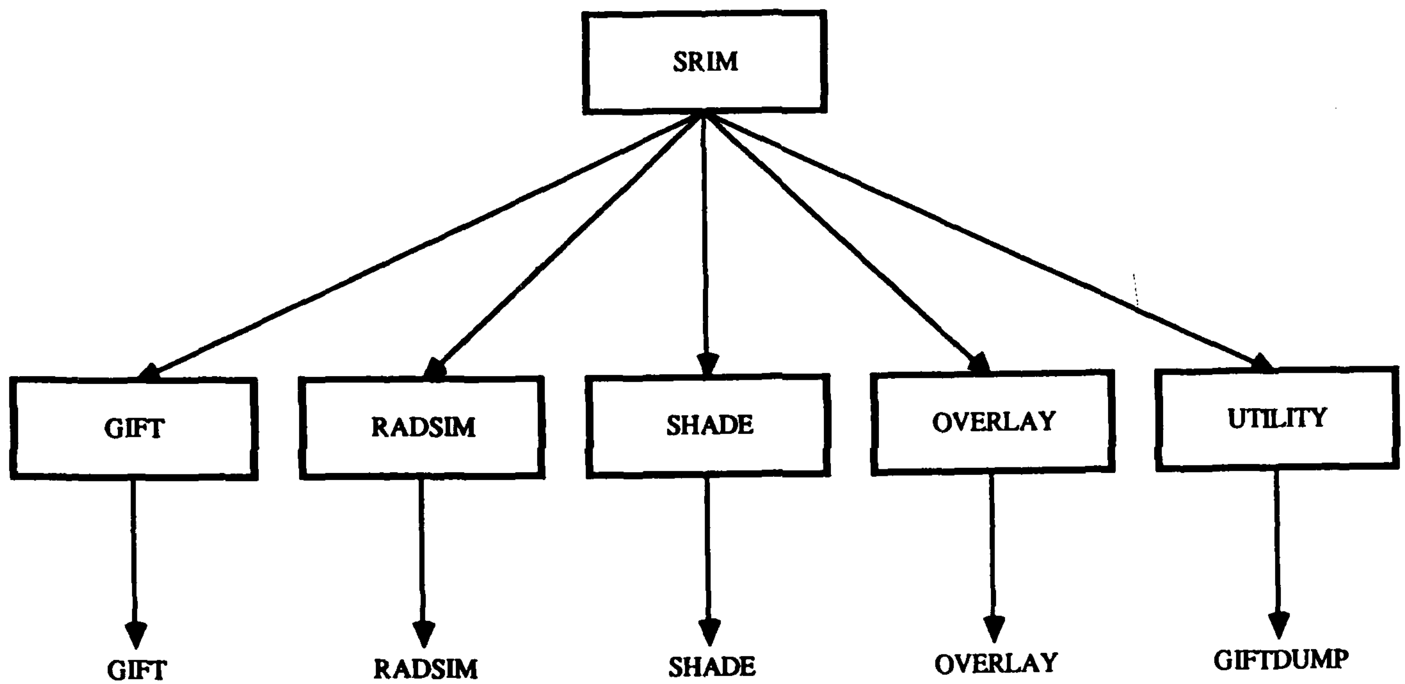


FIGURE 3-1. SRIM FILE STRUCTURE

```
Enter geometry file name: ? test.cg

GIFT Program - VAX/VMS FORTRAN 77 Version
Program set  1985 February 12
Execution date -   3-MAR-86


Input for Main
s - Print solids      r - Print Regions
i - Print Idents      a - Print Aster array
o - Print ordered id  e - Print ordered reg rpps
t - Overlap tol       l - LOS Tolerance
m - Max errors (25)
Main Option (End=0)?
```

FIGURE 3-2. GIFT STARTUP MESSAGE AND OPTIONS LIST.

```
Input for Main
s - Print solids       r - Print Regions
i - Print Idents       a - Print Aster array
o - Print ordered id   e - Print ordered reg rpps
t - Overlap tol        l - LOS Tolerance
m - Max errors (25)
Main Option (End=0)? s
Main Option (End=0)? 0

Enter geni

Title - sphere for tubular tests 2-1-85
Target units (in)

Number of solids     1
Number of regions    1


Description of Solids
  1    1sph       0.00      1.00      0.00      1.00

Location of solid pointers      lbody  =         1
Location of solid data          lsolid =         3

       rpp  box  sph  rcc  rec  trc  ell  raw  arb  tec  tgc  haf  tor  ars
Number    0    0    1    0    0    0    0    0    0    0    0    0    0    0
Storage   0    0    4    0    0    0    0    0    0    0    0    0    0    0

Storage for solid data                          19
Storage for solid pointers                       1
Total storage for solids                        20


** Diagnostic ** The following regions are null


Working Storage
Location of region ray storage  lnry   =        43
Location of rin  storage        lrin   =        44
Location of rout storage        lrot   =        45
Location of surfaces/ray num    lio    =        46

Loc next available storage      legeom =        47
```

FIGURE 3-3. OUTPUT FROM THE "PRINT SOLIDS" OPTION.

```
Input for Main
s - Print solids       r - Print Regions
i - Print Idents       a - Print Aster array
o - Print ordered id   e - Print ordered reg rpps
t - Overlap tol        l - LOS Tolerance
m - Max errors (25)
Main Option (End=0)? r
Main Option (End=0)? 0

Enter geni

Title - sphere for tubular tests 2-1-85
Target units (in)

Number of solids     1
Number of regions    1


Region combination data

 1   1    1    0    0    0    0    0    0    0    0

Location of region pointers      lregd  =       22
Location of region list          lregl  =       23

Number of descriptors                            1
Storage for region pointers                      1

Total storage for regions                        2


Region RPP equivalents

Region       xmin       xmax       ymin       ymax       zmin       zmax

   1       -1.0000     1.0000     0.0000     2.0000    -1.0000     1.0000

Location of region rpp equiv    lregmm =       24
Location of region list         lregon =       30
Index for middle of list        middle =        1
Total storage for region min and max   =       12


Enclosing  RPP
       xmin         xmax        ymin        ymax        zmin        zmax
     -1.0000       1.0000      0.0000      2.0000     -1.0000      1.0000

Location of enclosing RPP        lenrpp =       36


Working Storage
Location of region ray storage  lnry   =      43
Location of rin  storage        lrin   =      44
Location of rout storage        lrot   =      45
Location of surfaces/ray num    lio    =      46

Loc next available storage      legeom =      47
```

FIGURE 3-4. OUTPUT FROM "PRINT REGIONS" OPTION.

the example target. The non-printing options allow the user to change the following GIFT parameters:

- t-› Change the overlap tolerance which defaults to .01.
- m-› Change the number of execution errors that will be allowed before GIFT aborts. The default is 25.
- l-› Change the line-of-sight (LOS) tolerance which defaults to .01.

If these options are selected, GIFT will prompt for the new parameter values. More information on these options is only available from BRL. GIFT is normally run without selecting any of these options except the s and r options which can help verify a new geometry description file.

After option selection is complete (by entering 0), GIFT prints the title and units from the geometry file header record followed by the number of solids (primitives) and number of regions in the target. It then outputs the debug information (if requested) and lists any regions which are null(regions containning no primitives). The message "The following regions are null" will appear whether there are any null regions or not. If there are null regions, their region numbers will be listed after the message. Null regions do not cause a problem unless they are null due to an error in the geometry file. However, the program will run more efficiently if there are no null regions.

GIFT now prints out storage and tolerance information and prompts the user for the application subroutine to be executed. This output and prompt are shown in Figure 3-5. The application routines that can be used are:

- radar - Perform a ray trace and write a ray history file for use by RADSIM.
- optic - Perform a ray trace and write a SHADED optical image file for diplay.
- brandx - Only used to aid further program development
- check - Only used to aid further program development.
- end - End the program.

The brandx and check subroutines are not used for production.

3.2.1 RADAR APPLICATION SUBROUTINE

If the radar application is selected, the user will be prompted for:

Number of views -
: This allows the user to run several ray traces, changing only the emanation plane normal (aspect) between views. Only one ray trace file is output with the views separated by file markers. For release 2.0, this must be one.

Minimum number of reflections -
: A ray will not be traced past this number of reflections. This is usually set to 5 or 6.

Run identifier -
: A user chosen integer that is placed in the ray history header records to help identify the ray

```
Enter geni

Title - sphere for tubular tests 2-1-85
Target units (in)

Number of solids     1
Number of regions    1


** Diagnostic ** The following regions are null


Total storage for geometry data              43
Total working storage                         4
Total storage in master-aster                46

Tolerance for overlap        tol    =      0.0100
Tolerance for line of sight  tollos =      0.0100

Processed data written on file test.4
Time for input processing    0.00 seconds
Leave geni

Application Subroutines
end     brandx  check   optic   radar
Application Subroutine?
```

FIGURE 3-5. GIFT OUTPUT FOR NO OPTIONS AND THE SUBROUTINE SELECTION PROMPT.

history.

Comment -

Up to 72 characters of text for the ray history header.

Ray trace file name -

This is the file name for the ray history file. The default name is the name of the input geometry file with a .ray extension.

Azimuth (degrees)

This is the azimuthal angle for the emanation plane normal as shown in Figure 2-6.

Elevation (degrees) -

This is the elevation angle for the emanation plane normal as shown in Figure 2-6.

Max horz cells -

Number of rays to fire in each row from the emanation rectangle defined below.

Max vert cells -

Number of rays to fire in each row from the emanation rectangle defined below.

Note in the above prompts that GIFT uses cells instead of rays. To understand the relationship, imagine that the emanation rectangle in Figure 2-7 is covered by 12 rectangles (cells) of dimension HS by VS and a ray is fired from the center of each cell. For RADSIM it is more natural to think in terms of the ray spacing. After these prompts, GIFT has the information to set up the emanation rectangle and determine the ray firing points. These prompts are shown in Figure 3-6.

GIFT will place a rectangular parallelepiped (RPP) with sides perpendicular to the global coordinate system axes about the target that just encloses the target. This establishes the minimum and maximum values for the target coordinates in the overall coordinate system as shown in Figure 3-7. The center of the RPP is labelled as the center of the target and the length of the sides of the RPP gives the target dimensions. This information is printed out under the heading of target parameters as shown in Figure 3-9.

Next, GIFT will establish the emanation plane and rectangle as shown in Figure 3-8. The emanation plane is placed so that the length of the normal vector (back off distance) is 1.1 times the length necessary for the emanation plane to just clear the target. An enclosing RPP is placed around the target. The minimum and maximum distances in the direction of the emanation plane normal determines the depth of the target. The projection of this RPP into the emanation plane establishes the emanation rectangle. The sides of the emanation rectangle are parallel to the emanation plane horizontal and vertical axes. The emanation plane coordinates of the corners of the rectangle determine the range (of coordinates) and length of the sides of the rectangle in the emanation plane. The center of the emanation rectangle is found in the emanation plane coordinate system and the horizontal and vertical cell

size (ray spacing) is determined. This information is printed under the heading of "View Plane" as shown in Figure 3-9.

Finally, GIFT prints out the number of ray cells and prompts

Do you r(estart, c(ontinue, or e(nd: ?

The user responds with one of the following characters:

r-› Restart the radar subroutine from the begining.
c-› Continue with the radar subroutine.
e-› End the radar subroutine.

If the answer is c, GIFT performs the ray trace and writes out the ray history file. While performing the ray trace, GIFT outputs some summary information as shown in Figure 3-10.

3.2.2 OPTIC APPLICATION SUBROUTINE

If the user requests the optic subroutine, GIFT will issue the prompts shown in Figure 3-11. GIFT will prompt the user for:

Optical image file name -
  Name for the SHADED image output file.
Azimuth (degrees)
  Azimuth angle for ray trace.
Elevation (degrees) -
  Elevation angle for ray trace.
Max horz cells -
  Number of rays in a row.
Max vert cells -
  Number of rays in a column.
Illumination azimuth (degrees) -
  Azimuth angle for vector to the illumination source.
Illumination elevation (degrees) -
  Elevation angle for vector to the illumination source.

GIFT then prints out the same target parameters, and view plane information that it prints from the radar subroutine. In addition, GIFT outputs the minimum, center, and maximum of the pixel intensity range as white, gray, and black. The shading method is described at the end of Section 2.2. The components of the unit vector to the illumination source are also printed. After printing the above information (as shown in Figure 3-12), the user is prompted with

Do you r(estart, c(ontinue, or e(end: ? The user responds with one of the following characters:

r-› Restart optic subroutine from the beginning.
c-› Continue with the optic subroutine.
e-› End the optic subroutine.

After a responce of c, the ray trace is performed and the image file is written.

After either the radar or optic subroutines have finished, GIFT repeats the prompt shown in Figure 3-2. The user can then chose another subroutine to execute.

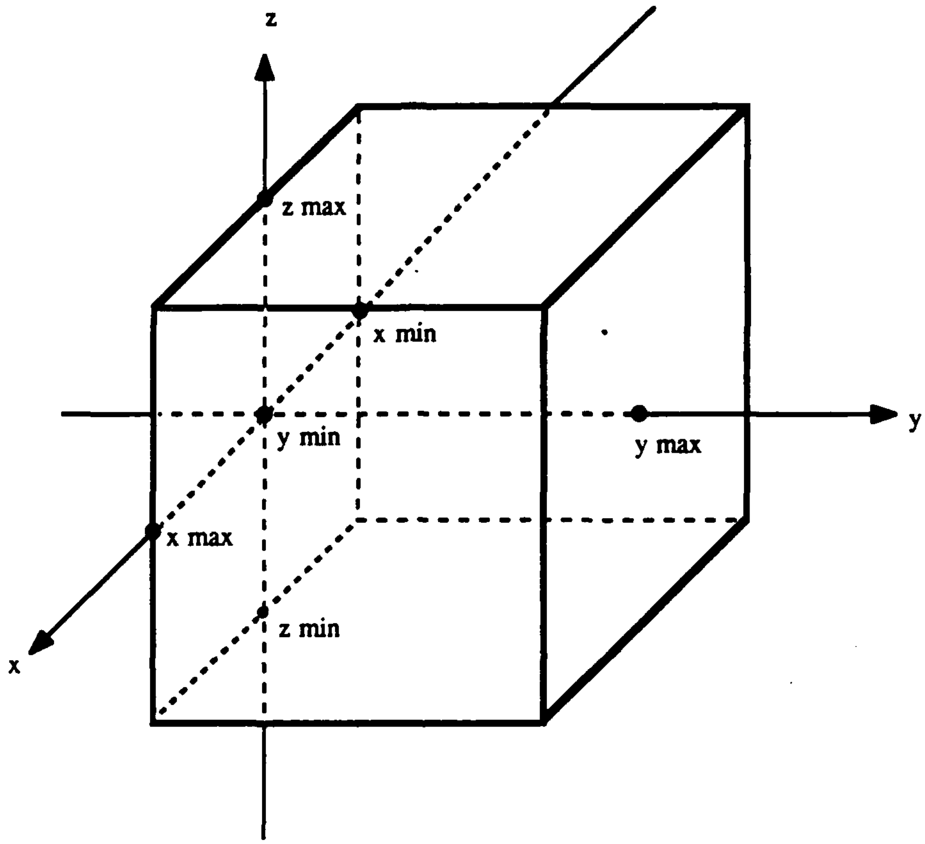
```
Application Subroutines
end     brandx  check   optic   radar
Application Subroutine? radar

Enter radar

radar version 07-feb-1985
modified to keep last ray
print range information
see George Darling about any problems
Bring hardcopy of results


Number of views? 1
Maximum number of reflections (.lt.255)? 2
run identifier? 1
Enter comment: test run for sphere
Enter ray trace file name: (<retn> = default) test.ray
Azimuth (degrees)? 0.
Elevation (degrees)? 30.
Max horz cells? 16
Max vert cells? 16
```

FIGURE 3-6. RADAR SUBROUTINE PROMPTS.



	min	max	center
x	-1	1	0
y	0	2	1
z	-1	1	0

FIGURE 3-7. ENCLOSING RPP FOR SPHERE OF RADIUS 1 CENTERED AT (0,1,0)

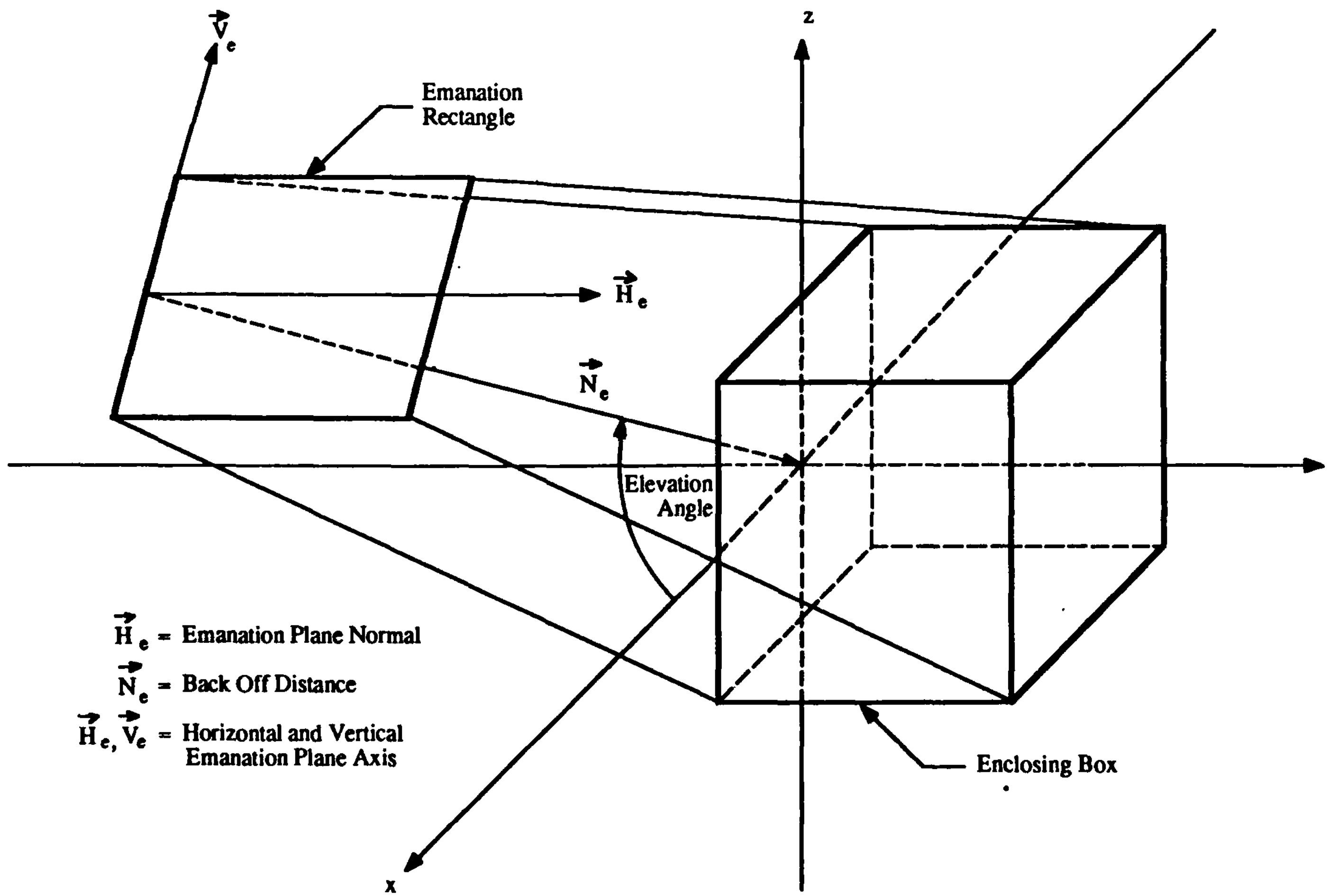


FIGURE 3-8. BOX FOR ESTABLISHING THE EMANATION RECTANGLE FOR ZERO AZIMUTH ANGLE

Target parameters	x	y	z
Minimum	-1.000	0.000	-1.000
Maximum	1.000	2.000	1.000
Center	0.000	1.000	0.000
Dimensions	2.000	2.000	2.000

View plane		
Horizontal length	2.000	
Vertical length	2.732	
Depth	2.732	
Back off distance	1.503	
Center	1.000	0.000
Horz Cell size	0.250	
Vert Cell size	0.342	
Horizontal range	0.000	2.000
Vertical range	-1.366	1.366

FIGURE 3-9. TARGET PARAMETER AND VIEW PLANE OUTPUTS.

```
Horz number cells          16
Number vert cells          16
Number of cells           256
Do you r(estart,c(continue or e(nd: ? c
the min and max ranges for 1st surfaces are:          0.104          0.947
The min and max total ranges are:                     0.104          0.947

Time for view    0.00 seconds

Time for radar    0.00 seconds

Leave radar
```

FIGURE 3-10. RADAR SUBROUTINE SUMMARY OUTPUTS.

```
Application Subroutines
end     brandx  check   optic   radar
Application Subroutine? optic

Enter optical

optical version 27-feb-1985
modified to keep last ray
print range information
see George Darling about any problems
Bring hardcopy of results


Enter optical image file name: (<retn> = default) test.mi
Azimuth (degrees)? 0.
Elevation (degrees)? 0.
Max horz cells? 16
Max vert cells? 16
illumination azimuth (degrees)? 0.
illumination elevation (degrees)? 45.
```

FIGURE 3-11. OPTIC SUBROUTINE PROMPTS.

```
Horz number cells          16
Number vert cells          16

Number of cells           256
Light source
  white                   255
  gray                    128
  black             0
  source direction      0.707     0.000     0.707
Do you r(estart,c(continue or e(nd: ? c

Time for optical     0.00 seconds

Leave optical
```

FIGURE 3-12. OPTIC SUBROUTINE SUMMARY OUTPUT.

3.3 GIFTDUMP

Figure 2-8 shows the inputs and outputs for the GIFTDUMP program while Figure 3-13 shows the terminal I/O for a run of GIFTDUMP. The run is for the ray history produced by the example GIFT run of Section 3.2. The run command for GIFTDUMP is

RUN [SRIM.UTIL]GIFTDUMP

After GIFTDUMP is started, it prompts the user for the name of the GIFT ray history file. The user enteres the name of the ray history file produced by a previous GIFT run. GIFTDUMP then outputs the ray history header information to the terminal and begins prompting for options. The GIFTDUMP options are:

Reflection histogram -

If the user requests a reflection histogram, GIFTDUMP will output the number of rays in the ray history having 0,1,2,...68 reflections. This output will be at the end of the output file.

Minimum number of reflections -

GIFTDUMP will only output the ray histories for rays with at least this number of reflections.

Specific records -

If the user wants to dump specific physical records, then GIFTDUMP will prompt for physical record numbers and only dump ray histories from these records.

These options are described in detail in the following paragraphs.

If the reflection histogram option is selected, GIFTDUMP must process the entire ray history file. This is done regardless of the specific record option. If the user is dumping only a few records from a large file, this will increase the time for the run.

A full dump of a ray history is usually very large. The size of the dump can be limited by use of the minimum number of reflections and specific record options. For example, if the user wishes to see if there are corner reflectors in the image, setting the minimum number of reflections to 3 will remove all of the rays from the dump that cannot have encountered a corner reflector. If the minimum number of rays is set to 1, then all rays that hit the target will be dumped.

If specific records are to be dumped, GIFTDUMP will prompt for the record numbers. After a record number is given, the record will be found and processed, then the user will be prompted for the next record number. The record numbers must be given in ascending order. A "return" for a record number prompt will terminate the record selection. If the histogram selection was not used, GIFTDUMP is finished. If the histogram option was selected, GIFTDUMP will continue to read through the ray history file to the end in order to complete the histogram information. In either case, the program will terminate with the end message shown in Figure 3-13 and the output will be in the file

```
        enter gift ray trace file name : test.ray
read record number   1
**** header information
        -1.00000     -1.00000      0.00000      1.00000      0.00000
         1.10000      0.00000      0.00000      2.00000      2.00000
        16.00000     16.00000      2.00000      1.00000

      test run for sphere

      run time: 16:45:51           date:  9-APR-85

do you want a reflection histogram? (y/n) : y
       minimum number of reflections (i) : 2
         do want specific records? (y/n) : y
    record number of record to dump (int) : 1
read record number   1
    record number of record to dump (int) : <CR>
read record number   2
read record number   3

normal termination
results are in giftdump.lis
FORTRAN STOP
```

FIGURE 3-13. EXAMPLE GIFTDUMP RUN.

GIFTDUMP.LIS which can then be printed.

If an error occures while GIFTDUMP is reading the ray history file, the output up to the point of the error will be in the output file and GIFTDUMP will end with an error message. Note that a histogram will not be output if an error occurs since it is not complete until all of the ray histories are read.

3.4 SHADE

The inputs and outputs for SHADE are shown in Figure 2-9 and the terminal I/O for a SHADE run is shown in Figure 3-14. The run in Figure 3-14 uses the GIFT ray history produced by the GIFT run in Section 3.2. The run command for SHADE is

RUN [SRIM.SHADE]SHADE

After SHADE is started, it prompts the user for the GIFT ray history file name. The user enters the name of a ray history file that has been produced by a previous run of GIFT. SHADE then outputs the ray history header information to the terminal and prompts for confirmation that this is the correct file. If the user answers "n", SHADE will terminate execution. If the answer is "y", SHADE will write the image size and the output record length to the terminal. The image size is determined by the number of rays fired in the GIFT run. In this case GIFT was run with 16 rays in a row (of horizontal cells) and 16 rows (of vertical cells). This gives an optical image size of 16X16 pixels and each record in the output file will contain a row of pixel intensities of 2 bytes each. This gives an output record length of 16*2 = 32 bytes. SHADE will now prompt the user for the name of the output optical image file.

Next SHADE will ask the user to define the direction to the source of illumination. If the answer to the prompt "Do you want default radar direction? (y/n):" is "y", then the reverse of the emanation plane normal is taken as the direction to the illumination source. If the answer is "n", the user is prompted for the azimuth and elevation angles for the vector from the center of the coordinate system to the illumination source. These angles are defined the same way the angles for the emanation plane normal were defined in Figure 2-6. After the illumination direction is given, SHADE outputs the three cartesian components of the illumination unit vector.

The output from SHADE is 16 bits per pixel. However, SHADE assumes a pixel word size of 8 bits and the intensities will be scaled to the range 0 to 255. SHADE outputs the maximum, middle, and minimum values of the intensity range as white, gray, and black. This completes the user inputs and SHADE proceeds to process the ray history file and output the optical image file. The procedure for calculating the pixel intensities is descibed in Section 2.4.

```
enter input ray trace file : test.ray


header record:
run id :         -1
view id:         -1
aim point:       0.00      1.00      0.00
dist:       1.10
elevation angle:      0.00
azimuthal angle:     0.000
emanation plane ht and wid:      2.00      2.00
ncols  :         16
nrows  :         16
maxref :          2


correct file? (y/n) :y


output image will have   16 rows and   16 cols
output file will have      32 bytes per record


enter output optical image file :test.ai


Do you want default radar direction? (y/n): n
Enter azimuth angle : 0.
Enter elevation angle : 60.


normalized light source:
           0.500000           0.000000           0.866025


white=      255 gray=      128 black=       0
```

FIGURE 3-14. EXAMPLE SHADE RUN.

3.5 OVERLAY

The inputs and outputs for OVERLAY are shown in Figure 2-10 and the terminal I/O for an example run is shown in Figure 3-15. The run command for OVERLAY is

RUN [SRIM.OVERLAY]OVERLAY

After OVERLAY is started, it prompts the user for the GIFT ray history file name. The user enters the name of a ray history file that has been produced by a previous run of GIFT. OVERLAY then outputs the ray history header information to the terminal and prompts for confirmation that this is the correct file. If the user answers "n", OVERLAY will terminate execution. If the answer is "y", OVERLAY will proceed.

OVERLAY now establishes the image and pixel sizes by prompting the user for the image size (in range and azimuth) and the pixel spacing. The image sizes are in pixels and the pixel spacing is the pixel size in distance units (feet, meters, etc.). The directions (rng,azim) are the slant range and azimuth directions as used by RADSIM (see Section 2.7.1). The pixel spacing is the pixel size in the slant range plane. For the example of Figure 3-15, the image will be 128 by 128 pixels with each pixel being .25 units on a side. The image will cover an area of 32 units by 32 units in the slant range plane.

The program now asks for range and azimuthal offsets. These offsets shift the target within the image. OVERLAY will set up the image so that the center of the overall coordinate system established by GIFT will be in the center of the image. If the target is not centered in this coordinate system, the user can give offsets that will bring the target to the center of the image. The offsets are given in distance units and move the target in the positive range and azimuth directions as defined in Figure 2-16. For example, suppose that the target is off center and the user wants to move it 10 pixels in range and -5 pixels in azimuth. The user would give offsets of (5.,-2.5) as shown in Figure 3-15 (since a pixel is 1/2 of a distance unit).

OVERLAY outputs 2 bytes for each pixel (an intensity value) and outputs the slant range optical image by row. Thus each record in the output file will contain 2*(number of pixels in a row) bytes. The image size was given by the user and a column corresponds to range while a row corresponds to azimuth. OVERLAY now writes the image and output record sizes to the terminal. In Figure 3-15, the image size is 128 by 128 pixels and the output records are 256 bytes. OVERLAY now prompts for the name of the output file.

Next OVERLAY will ask the user to define the direction to the source of illumination. If the answer to the prompt "Do you want default radar direction? (y/n):" is "y", then the reverse of the emanation plane normal is taken as the direction to the illumination source. If the answer is "n", the user is prompted for the azimuth and elevation angles for the vectors from the center of the coordinate system to

```
enter input ray trace file : test.ray


header record:
run id :         -1
view id:         -1
aim point:       0.00      1.00      0.00
dist:      1.10
elevation angle:       0.00
azimuthal angle:      0.000
emanation plane ht and wid:       2.00      2.00
ncols  :         16
nrows  :         16
maxref :          2


correct file? (y/n) : y


Enter image range size : 128
Enter image azimuth size : 128
Enter pixel spacing : .25


Enter the range offset : 0.
Enter the azimuth offset : 0.


output image will have  128 rows and  128 cols
output file will have      256 bytes per record


enter output optical image file : test.mi


Do you want default radar direction? (y/n): n
Enter azimuth angle 0.
Enter elevation angle 45.


normalized light source:
           0.707107            0.000000            0.707107


white=     255. gray=     128. black=       0.
$
```

FIGURE 3-15. EXAMPLE OVERLAY RUN.

the illumination source. These angles are defined the same way the angles for the emanation plane normal were defined in Figure 2-6. After the illumination direction is given, OVERLAY outputs the three cartesian components of the illumination unit vector.

OVERLAY assumes a pixel word size of 8 bits and the image pixels will be to be in the integer range 0 to 255. The method used to compute the intensity of a return is described in Section 2.5. OVERLAY now has all the information it needs and processes the ray history. The result will be a slant range optical image file.

3.6 DETECT

The inputs and outputs for the DETECT program are shown in Figure 2-12 and the terminal I/O for a DETECT run is shown in Figure 3-16. The run uses a complex image file generated by the RADSIM run illustrated in Section 3.7. The run command for DETECT is

RUN [SRIM.DETECT]DETECT

When DETECT starts, it will output its startup message and prompt the user for a command. The command prompt sequence starts by printing three lines of information for the user:

- "Complex image file" -> This is the name of the input complex image file. This will be a blank until a file name is given.
- "Complex image maximum magnitude" -> When DETECT reads in a complex image file, it finds the magnitudes of the complex pixels. This is the largest of these pixel magnitudes. This is zero until a complex image file has been read in.
- "User specified scaling factor" or "Autoscaling scaling" -> If the user has specified the scaling factor then the first of these messages will appear. If the user has asked for automatic scaling, then the second message will appear. The first message is given with a scaling factor of zero until a complex image has been read in.

The allowed commands that be given in response to this prompt are:

- f-> Set input file. DETECT will prompt the user for the name of the complex image file to be processed.
- s-> Set scale. DETECT will prompt the user "(A)utoscale or (M)anual? :". If the user enters "a", DETECT will determine the scaling factor so that the detected image pixels will contain integers in the range 0 to 255. If the user enters "m", DETECT will prompt the user for a scale factor. This scale factor must be chosen so that the product of the "Complex image maximum magnitude" and the scale factor will be less than 32,768 (largest signed integer for 16 bit words).
- d-> Detect image. DETECT will prompt the user for the

```
***SRIM detection program***


Complex image file :
Complex image maximum magnitude :    0.000000E+00
User specified scaling factor :      0.000000E+00

Commands :
 [F] - set input file
 [S] - set scale
 [D] - detect image
 [E] - exit program

Enter command : f
Enter input filename : test.ci

Complex image file :             test.ci
Complex image maximum magnitude :    0.142081E+01
User specified scaling factor :      0.000000E+00

Commands :
 [F] - set input file
 [S] - set scale
 [D] - detect image
 [E] - exit program

Enter command : s
(A)utoscale or (M)anual? : a

Complex image file :             test.ci
Complex image maximum magnitude :    0.142081E+01
Autoscaling factor :                 0.179475E+03
```

FIGURE 3-16. EXAMPLE DETECT RUN.

```
  run [srim.radsim]radsim
single precision eps =          0.22204460E-15          = 2(- 52)
double precision eps =      0.516987882845642D-25       = 2(- 84)

enter ray history file name:test.ray

ay history file header record:
               -1 run id         -1 view id
           0.000              1.000              0.000          aim point
           1.100 dist         0.000 el           0.000 az
           2.000 ht           2.000 wd
              16 nrow            16 ncol
               2 maxref           1 shot

correct ray history file? [y/n]y

vertical emanation vector =     0.000000E+00  0.000000E+00  -0.100000E+01
horizontal emanation vector =   0.000000E+00  -0.100000E+01  0.000000E+00
normal vector =                -0.100000E+01  0.000000E+00   0.000000E+00
```

FIGURE 3-17. RADSIM STARTUP AND RAY HISTORY FILE SELECTION.

continue. If the user answers "n", RADSIM will ask if the user wishes to continue. A "n" will terminate the program while a "y" will cause RADSIM to repeat the prompt for the name of the surface type file.

After the surface type file is opened and verified, RADSIM will prompt for the name of the radar file. This file defines the radar system for RADSIM and is described in detail in Appendix G. After opening the radar file, RADSIM lists its contents to the terminal and asks if this is the correct file. Note that a "carriage return" for the file name prompt causes the radar file name to default to radar.dat. This is shown in Figure 3-19. If the user answers "y", RADSIM will continue. If the user answers "n", RADSIM will ask if the user wishes to continue. A "n" will terminate the program while a "y" will cause RADSIM to repeat the prompt for the radar file name. In this case the radar is an x-band radar (wavelength of .1 foot) with a resolution of 5 feet. It transmitts and receives vertical polarization and has a Taylor weighted system response. The image will be sampled twice per resolution length (pixel size is 2.5 feet). All of the radar file entries are covered in detail in Appendix G.

RADSIM now prompts the user for image information as shown in Figure 3-20. First the user is prompted to select the image type. RADSIM can output either a scaled magnitude or a complex image. If the scaled magnitude image is selected, RADSIM will output a detected image with the pixel values scaled to the integer interval 0 to 255 (pixel word size of 8 bits). Next RADSIM prompts for the image size and the offsets. The image size is given in pixels. In this example, the image size is given as 64 by 64. Since a pixel is 2.5 feet on a side, this gives a coverage of 160 ft. by 160 ft. in the slant range plane. The range and azimuth directions are the directions shown in Figure 2-16. If the image size in the range or azimuth direction is not 64,128,256, or 512, RADSIM flags an error and reprompts for the image size. The offsets shift the target within the image. RADSIM will set up the image so that the center of the overall coordinate system established by GIFT will be in the center of the image. If the target is not centered in this coordinate system, the user can give offsets that will bring the target to the center of the image. The offsets are given in distance units and move the target in the positive range and azimuth directions as defined in Figure 2-16. If the user wishes to reposition the target in the image by 10 feet in range but not reposition it in azimuth, they would give the offsets as 10.,0.. This would move the target 4 pixels (10/2.5) in the +range direction. In this example, the offsets are zero. After the offsets have been given, RADSIM outputs the following information to the terminal.

scale = The number of pixels per resolution. In this example the scale is two. Thus the image is being sampled twice per resolution.

azthr = The azimuth of a return must be greater than

```
Commands :
 [F] - set input file
 [S] - set scale
 [D] - detect image
 [E] - exit program

Enter command : s
(A)utoscale or (M)anual? : m
Enter scale factor : 512

Complex image file :              test.ci
Complex image maximum magnitude :    0.142081E+01
User specified scaling factor :      0.512000E+03

Commands :
 [F] - set input file
 [S] - set scale
 [D] - detect image
 [E] - exit program

Enter command : d
Enter output filename : test.mi

Complex image file :              test.ci
Complex image maximum magnitude :    0.142081E+01
User specified scaling factor :      0.512000E+03

Commands :
 [F] - set input file
 [S] - set scale
 [D] - detect image
 [E] - exit program

Enter command : e
$
```

FIGURE 3-16. EXAMPLE DETECT RUN.
(Concluded)

name of the detected image output file. It will then perform the detection (as described in Section 2.6) and write out the detected image to the specified file.

e-› exit program. DETECT will terminate execution.

When running DETECT, the usual sequence of commands would be "set input file", "set scale", "detect image", "and "end". However, the commands can be given in any order with the restriction that the first command must be a "set input file" command and a scale factor must be set before the "detect image" command is given. In the example run of Figure 3-16, the scaling factor was first set by autoscaling and then reset by a user given value. Another example would be to follow the "detect image" command with a "set input file" command and begin processing another complex image file.

3.7 RADSIM

The inputs and outputs for RADSIM are shown in Figure 2-13. Running RADSIM will be illustrated by using the ray history file produced in the example GIFT run of Section 3.2. The target was a sphere of radius 1 centered at (0,1,0) in the overall coordinate system and the units used in the example are feet. The GIFT run fired a 16 by 16 grid of rays (256 rays). The run command for RADSIM is

RUN [SRIM.RADSIM]RADSIM

After RADSIM has started, it determines the parameter values it will use for single and double precision accuracy. These values are used as tolerances for floating point number comparisons (for example, if a variable is zero). RADSIM will then prompt the user for the name of the GIFT ray history file to use as its geometry input as shown in Figure 3-17. After the user supplies the file name, RADSIM outputs the ray history header information to the terminal and ask if this is the correct GIFT file. If the user answers "y", RADSIM will continue. If the user answers "n", RADSIM will ask if the user wishes to continue. An "n" will terminate the program while a "y" will cause RADSIM to repeat the prompt for the ray history file name. Once the correct file is opened and verified, RADSIM prints the emanation plane vertical unit vector, horizontal unit vector, and unit normal. The terminal I/O up to this point is shown in Figure 3-17.

RADSIM will now prompt for the name of the surface type file. This is the file that associates surfaces in the target with radar reflectivity models and is described in detail in Appendix G. The contents of the surface file are listed to the terminal as shown in Figure 3-18. In this case the file contains one surface which is assigned reflection model number one. The header information helps identify the surface file as the correct one for this run. The user is asked to confirm that this is the correct surface type file. If the user answers "y", RADSIM will

```
Enter surface type file name : test.sur

Surface parameters are:
numsur =     1
index     region     body type     surface type     model number
   1         1           1              1                1

Correct surface file type? [y/n] : y
```

FIGURE 3-18. SURFACE TYPE FILE SELECTION IN RADSIM.

```
Enter radar file name
   [default is 'radar.dat'] :<CR>

WAVLEN                    0.1
RESOL                     5.000000
XVPOL                     1.000000
XHPOL                     0.000000
RVPOL                     1.000000
RHPOL                     0.000000
SPACEN                    2.500000
NOIS                      0.000000
SLL                     -30.000000
IPRSIZ                    0.000000
NTAYLOR                   1.000000
CONV. CELLS              11.000000
QUANT. FACT.             10.000000
FRACT BW                  0.392000
11 pixels will be used for range convolution
correct file? [y/n] :y
```

FIGURE 3-19. RADAR FILE SELECTION IN RADSIM.

```
scaled magnitude or complex image ? [m/c] m
enter image size: [2i5: range,azimuth]64,64
enter range and azimuth offsets: [2e10.5]<CR>

   scale=    0.20000E+01
   azthr=   -0.80000E+02
 halfrng=    0.80000E+02
  center=    0.11000E+01

enter output image file name: test.mi
create new file or attach existent? (c/a):c
```

FIGURE 3-20. IMAGE SPECIFICATION IN RADSIM.

this number for the return to be in the image. In the example run the image spans -16 feet to +16 feet in azimuth. The azthr value is changed by the azimuth offset.

halfrng = One half of the range spanned by the image. Returns with ranges between (center - halfrng) and (center + halfrng) will be in the image.

center = Range (measured from the emanation plane) of a return that will be at the center of the image. In the example, a return with a range of 1.1 feet will be at the center of the image (in range). This is the back off distance plus the range offset.

These numbers for the example image are shown in Figure 3-20.

Now RADSIM will prompt the user for the name of the output image file. Once the file name is given, RADSIM asks whether the user wants to create a new file or attach an existing one. If the answer is "c", RADSIM will create a new image file with the specified name. If the answer is "a", RADSIM will open an existing file with the specified name. If the file is attached it must be a complex image file. RADSIM will initialize the complex image by reading in this file and then will process the ray history file specified earlier. The returns from the ray history processing will be coherently added to the existing image in the file. The file will then be rewritten with the new data at the end of the run. These prompts are shown in Figure 3-20.

RADSIM next prompts the user for the name of the reflection model file. This is the file that defines the reflection models that will be associated with the target surfaces. If the user responds to the prompt with a "carriage return", the file name will default to surface.dat. The number of reflection models defined in the file, the number of these models that have been referenced in the surface type file, and the data defining the reflection models are output to the terminal. The contents of the reflection model file are described in Appendix G. The user is then prompted to verify that the correct file has been opened. If the user answers "y", RADSIM will continue. If the user answers "n", RADSIM will ask if the user wishes to continue. A "n" will terminate the program while a "y" will cause RADSIM to repeat the prompt for the reflection model file. The reflection model I/O from RADSIM is shown in Figure 3-21.

RADSIM now prompts the user for zones of interest. If the user responds "n", RADSIM begins processing the ray history file. If the user responds "y", RADSIM begins the series of prompts shown in Figure 3-22. Zones of interest are defined by giving min. and max. pixel ranges in range and azimuth that define a rectangle in the image. If a return comes back in a zone of interest, information about this return is output to the user. This can be very useful for analysing an image. The user can define up to 10

different zones of interest in the image. If the user asks for zones of interest, RADSIM then asks whether the user wants the output to come to the terminal or be written to a file for printing. If the user responds "t", the output will be written to the terminal. If the user responds "l", the output will be written to a disk file called zones.dat which can be printed after the run finishes. After selecting the output option, RADSIM will prompt the user for the pixel ranges defining the zones of interest. After the zones of interest are defined RADSIM begins processing the ray history file.

During processing, if a return is in a zone of interest, RADSIM prints out the information shown in Figure 3-22. Each return into a zone of interest causes the following output:

rayout -> Labels output as comming from the RAYOUT subroutine

ipix -> The pixel number within a row that the return is in. This corresponds to the range position in the image.

irec -> The row (or record number in the image file) that the return is in. This corresponds to the azimuth position in the image.

fi -> The real part of the complex amplitude for the return

fq -> The imaginary part of the complex amplitude for the return.

dist -> The range position in distance units for the return. In the example this distance is in feet.

azimu -> The azimuth position in distance units. In the example this is in feet.

nrec -> The record number of the physical record in the ray history containing the ray that caused the return.

rayptr -> Pointer to the firing record within the physical record that starts the ray history for the ray causing the return.

refptr -> Pointer to the reflection record within the physical record for the reflection causing the return.

For example, the first return in the zone of interest has the image location (32,32). The complex amplitude is (-0.14573e-01,.77816e-01). The coordinates of the return are range = .12 feet, azimuth = .937 feet. Finaly, the ray history for the ray causing the return is in physical record number 1 and starts at logical record 241 (the firing record) while the reflection record for the return is logical record 242.

After RADSIM is finished, it outputs summary information to the terminal as shown in Figure 3-22. This information is:

number records -> The number of physical records that RADSIM read from the ray history file.

totali -> The real part of the total complex return for the target

```
Enter reflection model file name
   [default is 'surface.dat'] :<CR>

Reflection model file contains   5 models.
Modtyp array references   1.
```

	MU	EPS	RHO	ROUGH	ABSORB	POLSCA	PHASCA
1	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
2	0.0000	0.0000	0.0000	0.0000	0.5000	0.0000	0.0000
3	0.0000	0.0000	0.0000	0.0000	1.0000	0.0000	0.0000
4	20.0000	15.0000	0.0000	4.0000	0.5000	0.0000	0.0000
5	5.0000	12.0000	6.0000	3.0000	0.0000	0.5000	1.0000

```
correct file? [y/n]:y
```

FIGURE 3-21. REFLECTION MODEL FILE SELECTION IN RADSIM.

totalq -› The imaginary part of the total complex return for the target.
rcs -› The radar cross section for the target = totali**2 + totalq**2. .
rtncnt -› The number of returns that make up the image.
image scale factor -› For a scaled magnitude image, this is
the number that is multiplied by the magnitude
of the complex image pixels to scale the image.
The result is an image whos pixels contain
an integer between 0 and 255.

```
define zones of interest in image? [y/n]y
send to lp or ti? [l/t]l
send to lp or ti? [l/t]t
how many zones? (1 to 10)1
enter rngmin, rngmax, azmin, azmax: (4i)
1,64,1,64
rayout ipix,irec,fi,fo =   32   32 -0.14573E-01  0.77816E-01
    dist,azimu,nrec =  0.120E+00  0.937E+00     1
     rayptr,refptr =   241  242
rayout ipix,irec,fi,fo =   32   32 -0.14610E-01  0.77729E-01
    dist,azimu,nrec =  0.120E+00  0.106E+01     1
     rayptr,refptr =   244  245
rayout ipix,irec,fi,fo =   32   32 -0.14547E-01  0.77884E-01
    dist,azimu,nrec =  0.120E+00  0.812E+00     2
     rayptr,refptr =    31   32
rayout ipix,irec,fi,fo =   32   32  0.36850E+00 -0.50559E-01
    dist,azimu,nrec =  0.104E+00  0.937E+00     2
     rayptr,refptr =    34   35
rayout ipix,irec,fi,fo =   32   32  0.36853E+00 -0.50319E-01
    dist,azimu,nrec =  0.104E+00  0.106E+01     2
     rayptr,refptr =    37   38
rayout ipix,irec,fi,fo =   32   32 -0.14656E-01  0.77620E-01
    dist,azimu,nrec =  0.120E+00  0.119E+01     2
     rayptr,refptr =    40   41
rayout ipix,irec,fi,fo =   32   32 -0.14557E-01  0.77860E-01
    dist,azimu,nrec =  0.120E+00  0.812E+00     2
     rayptr,refptr =    79   80
rayout ipix,irec,fi,fo =   32   32  0.36851E+00 -0.50[illegible]6E-01
    dist,azimu,nrec =  0.104E+00  0.937E+00     2
     rayptr,refptr =    82   83
rayout ipix,irec,fi,fo =   32   32  0.36854E+00 -0.50256E-01
    dist,azimu,nrec =  0.104E+00  0.106E+01     2
     rayptr,refptr =    85   86
rayout ipix,irec,fi,fo =   32   32 -0.14665E-01  0.77597E-01
    dist,azimu,nrec =  0.120E+00  0.119E+01     2
     rayptr,refptr =    88   89
rayout ipix,irec,fi,fo =   32   32 -0.14602E-01  0.77747E-01
    dist,azimu,nrec =  0.120E+00  0.937E+00     2
     rayptr,refptr =   130  131
rayout ipix,irec,fi,fo =   32   32 -0.14638E-01  0.77661E-01
    dist,azimu,nrec =  0.120E+00  0.106E+01     2
     rayptr,refptr =   133  134
done number records =           3
 totali =  0.136E+01  totala =  0.420E+00  rcs =  0.202E+01  [illegible]tncnt =      12
 image scale factor =    0.191744E+03
FORTRAN STOP
```

FIGURE 3-22. ZONES OF INTEREST AND ENDING INFORMATION FOR RADSIM.

4 SETTING UP SRIM RUNS

4.1 FILE SIZES AND RUN TIMES

The three largest files generated by the SRIM programs are the ray history file, the complex image file, and the detected image file. The ray history file can be large enough to limit the number of rays that can be fired. The contents of the ray history file are described in Appendix B.

Each logical record in the ray history file contains 18 four byte words or 72 bytes of information. The physical records are fixed length records containing 256 logical records or 18,432 bytes. Since a ray history cannot cross a physical record boundary, not all of the 256 logical records will contain real information so there is some wasted space. Typical physical records will contain 250 to 256 logical records and only 2 or 3 percent of the file space is wasted. Each ray in a ray history consists of a firing record, a reflection record for each reflection, and an escape record. Thus a ray has 2 + N logical records in the ray history where N is the number of reflections for the ray. The first three logical records in the ray history contain header information. This gives a file size of

```
     file size = 72*(2*NRAY + N + 3)
     where:
            72 = number of bytes in a logical record.
          NRAY = number of rays that hit the target.
             N = number of reflections for all NRAY rays.
               = NRAY*NAVE
          NAVE = Average number of reflections for a
                 ray that hits the target.
             3 = Number of header logical records.
```

As an example, suppose that a 256 by 256 grid of rays is to be fired by GIFT. It is estimated that 80% of the rays will hit the target and that NAVE is 1.7. Then

```
          NRAY = 256*256*.8 = 52,429
             N = 52,429*1.7 = 89,129
     file size = 72*(2*52,429 + 89,129 + 3)
               = 13,967,280 bytes
```

If a fudge factor is included for errors in NAVE and NRAY as well as for wasted space, a reasonable estimate of the ray history file size is approximately 15 Mbytes or 30,000 blocks on the VAX/780 disk.

The size of the image files is much simpler. The complex image file contains two real numbers for each pixel or 8 bytes per pixel while the detected image contains a 16 bit integer for each pixel or 2 bytes per pixel. This gives file sizes of:

```
      complex image file size = 8*RNGPIX*AZPIX
     detected image file size = 2*RNGPIX*AZPIX
     where:
```

RNGPIX = number of pixels in range direction
AZPIX = number of pixels in azimuth direction

For example, a 128 by 128 pixel image would result in the following file sizes:

complex image file size = 131,072 bytes
detected image file size = 32,768 bytes

These are small compared to the size of a typical ray history file. However, the complex image file for a 512 by 512 image would be 2 Mbytes which can add up if several images are run. The file sizes of SHADE and OVERLAY images are the same as the detected image file size while RADSIM can produce either complex or detected image files.

The two programs that have (by far) the largest run times are GIFT and RADSIM. The run time for GIFT depends on the number of regions, the region definitions, and the number of rays that are fired. These factors effect both the optic and radar subroutines. The optic subroutine will run faster on the same geometry and number of rays because it only tracks a ray to the first reflection and does not output a ray history file. Further, for quick optical images, a small number of rays can be fired. A crude but useful optical image can be produced with 64 by 64 to 128 b 128 grids of rays which is smaller than the number of rays usually required for a radar image.

Every time GIFT attempts to find a reflection point for a ray, it must look through the list of regions to find which region the ray intersects first. When GIFT determins if a region reflects the ray, it must evaluate the Boolean operations that were used to define the region with the primitive solids. These considerations lead to two rules for speeding up GIFT:

1. Build the model of the target with the minimum number of regions and primitives that are required to describe the target with acceptable accuracy.
2. Keep the region definitions as simple as possible. One primitive per region is ideal.

For radar simulations, small features can sometimes be omitted from the model since they have a small effect on the image. The number of rays and the maximum number of reflections allowed for a ray are usually determined by the nature of the target and radar.

The run time for RADSIM depends on the size of the ray history file (the number of rays and reflections), the number of pixels used in the system response convolution, and the use of additive noise. The size of the ray history file is a measure of the number of returns that must be computed and the run time will increase linearly with an increase in the number of returns. Each return is convolved with the system response (in the range direction). The more output pixels used for the convolution, the longer it will take to compute the convolved return. When additive noise is included in a run, RADSIM initializes each pixel in the complex image to a pair (real,imaginary) of Gaussian random numbers. For a 128 by 128 image, RADSIM will generate 32,768 random numbers. This will add significantly to the

run time.

For targets with a large number of bodies (several hundred or more), GIFT runs will generally take longer than RADSIM runs while targets with a small number of bodies (less than 50 or so), it will usually take longer to run RADSIM than GIFT. Figure 4-1 shows the CPU times for runs of the SRIM programs for a specific case. The target used is the DICYII object which is discussed in section 4.5 and shown in Figure 4-8. all runs were done with 128 by 128 ray grids.

4.2 CHOOSING THE RAY SPACING IN GIFT

The ray spacing in GIFT depends on the use (optical or radar image) that the ray trace is put. For optical images, the ray density is chosen to show the target features and control the blockyness of the image. Each ray represents a pixel in the image and fewer rays correspond to larger (in space dimensions) blocks being used to represent the image. This is similar to grain size in photographic film. If a fine grain image is desired, a large number of rays must be fired. If a low number of rays is used, the image must usually be magnified to obtain a proper image size and this will result in a blocky appearance.

If the ray trace is being done to generate a radar image, then RADSIM's requirements must be used to determine the ray spacing. The ray spacing for RADSIM is driven by four requirements:

1. The geometry of the target must be properly sampled.
2. The system response of the radar must be properly sampled.
3. The phase variation of the physical optics returns must be properly sampled.
4. The number of returns per pixel must show a smooth variation over the image.

The ray spacing is chosen by determining the maximum spacing allowed by each of the RADSIM requirements and setting the ray spacing to the minimum of these spacings.

For geometric sampling, the ray spacing must be fine enough to define the multiple reflection geometry and the stationary phase regions. A simple criteria is that the ray trace should be fine enough to produce a good (not blocky) optical image. This will give good geometric sampling for the first reflection. Multiple reflection criteria are more difficult. Consider the target shown in Figure 4-2. The example rays shown in the figure illustrate the divergence introduced by the rays being reflected from a curved surface. If good sampling is desired for the second reflections, the ray spacing must be smaller than an optical image criteria would indicate. The spacing of the rays at the second reflection is a function of the divergence caused by the first reflection and the distance from the first reflection to the second reflection. The ray spacing for sampling a multiply reflecting geometry is a question of

Run times are for the DICYII target with 128 by 128 ray traces.
The runs are shown in the work sessions (Appendices C-F, I).

Program	CPU Time
GIFT (optic)	28.5 sec
GIFT (radar)	1 min, 7.6 sec
SHADE	5.25 sec
OVERLAY	5.91sec
RADSIM	1 min, 28.2 sec
GIFTDUMP	8.62 sec
DETECT	3.42 sec

FIGURE 4-1. EXAMPLE CPU TIMES FOR SRIM PROGRAMS

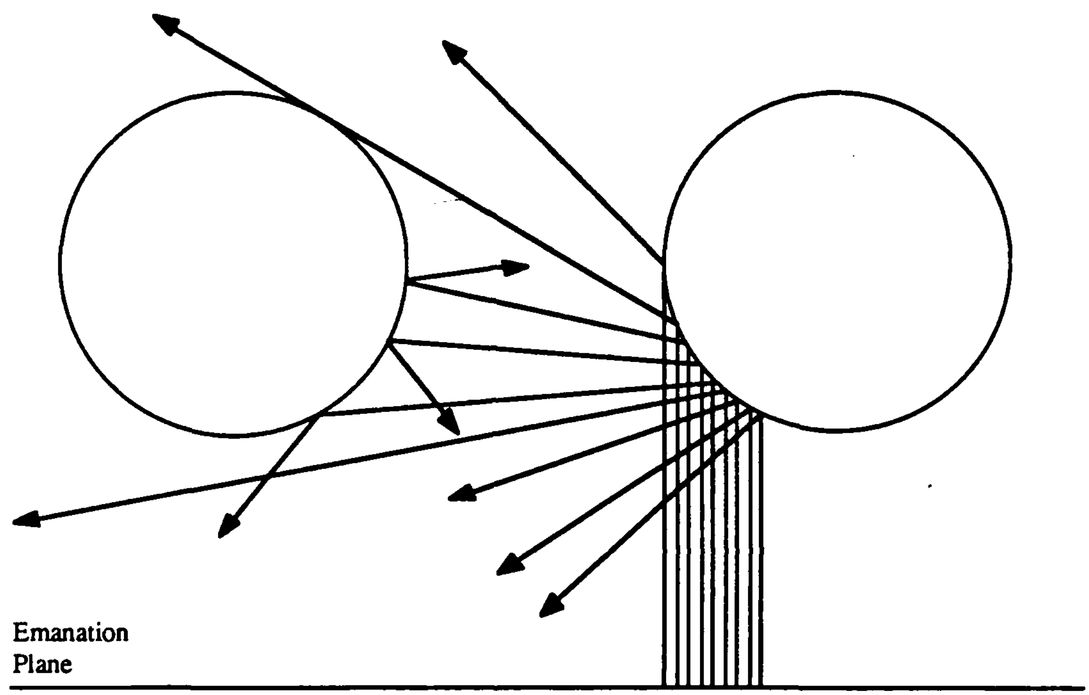


FIGURE 4-2. ILLUSTRATION OF RAY DIVERGENCE.

judgment that is not easily quantified. The factors to be considered are the curvatures of the target surfaces, the distances between surfaces, and the maximum number of reflections along a ray that are considered important.

The stationary phase regions are due to the interaction between the target geometry and the radar wavelength. The shorter the wavelength, the smaller the stationary phase regions will be. Examples of stationary phase regions are shown in Figure 2-17. The characteristic dimension of a stationary phase region as a function of wavelength and radius of curvature is shown in Figure 4-3. For example, a sphere of radius 2 feet will have a stationary phase region for x-band (wavelength = .1 feet) of radius .223 feet. For good results, the ray spacing should be smaller than 1/2 of the characteristic dimension of the smallest stationary phase region in the target.

The system response is applied to each ray in the range direction and to each pixel in the azimuthal direction. Figure 4-4 shows how the sampling of the system response can effect the image. Figure 4-4a is the reflectivity that would result with an infinitesimal ray spacing (an analogue system). The returns in RADSIM are a discrete sampling of this reflectivity. The effect of a ray spacing that is too course is shown in Figure 4-4b while 4-4c shows the result of a properly sampled system response. The ray spacing in Figure 4-4c is small compared to the resolution of the radar and produces a smooth complex image. A ray spacing that is less than 1/5 of the resolution will produce a reasonably smooth image.

For curved surfaces, the effects of phase cancelling due to range variation is handled by defining a stationary phase region. For flat surfaces, this cannot be done. The phase cancellation over a flat surface must be obtained by proper ray sampling of the phase. Figure 4-5 shows a flat plate whos normal makes an angle of A degrees with the emanation plane normal (radar look direction). Two rays are shown which are separated by the ray spacing in the vertical direction. The returns from these rays will differ in phase by

del phase = 4*PI*d*tan(A)/lamda

which represents a sampling of the phase ramp shown to the left in Figure 4-5. The slope of the phase ramp is

slope = 4*pi*x*tan(A)/lamda

and to correctly reconstruct the return from the plate, this ramp must be sampled at intervals of < PI radians (Nyquist). In terms of the ray spacing d, this requires that

del phase < PI

or d < lamda/(4*tan(A))

The table in Figure 4-6 shows the maximum allowed ray spacing for different values of A with a wavelength of .1 feet (x-band). It is clear that for any ray spacing, there is an angle above which the phase ramp will be undersampled. It is sometimes not possible to meet this ray spacing requirement. Fortunatly, an acceptable image can often result from an undersampled phase variation. This can

R \ λ	.01	.1	1.0
2	.071	.223	.696
4	.100	.316	.992
6	.122	.387	1.218
8	.141	.447	1.409
10	.158	.500	1.576

R = Radius of Curvature in Feet

λ = Wavelength in Feet

Table Entries are R_c = Characteristic Dimension

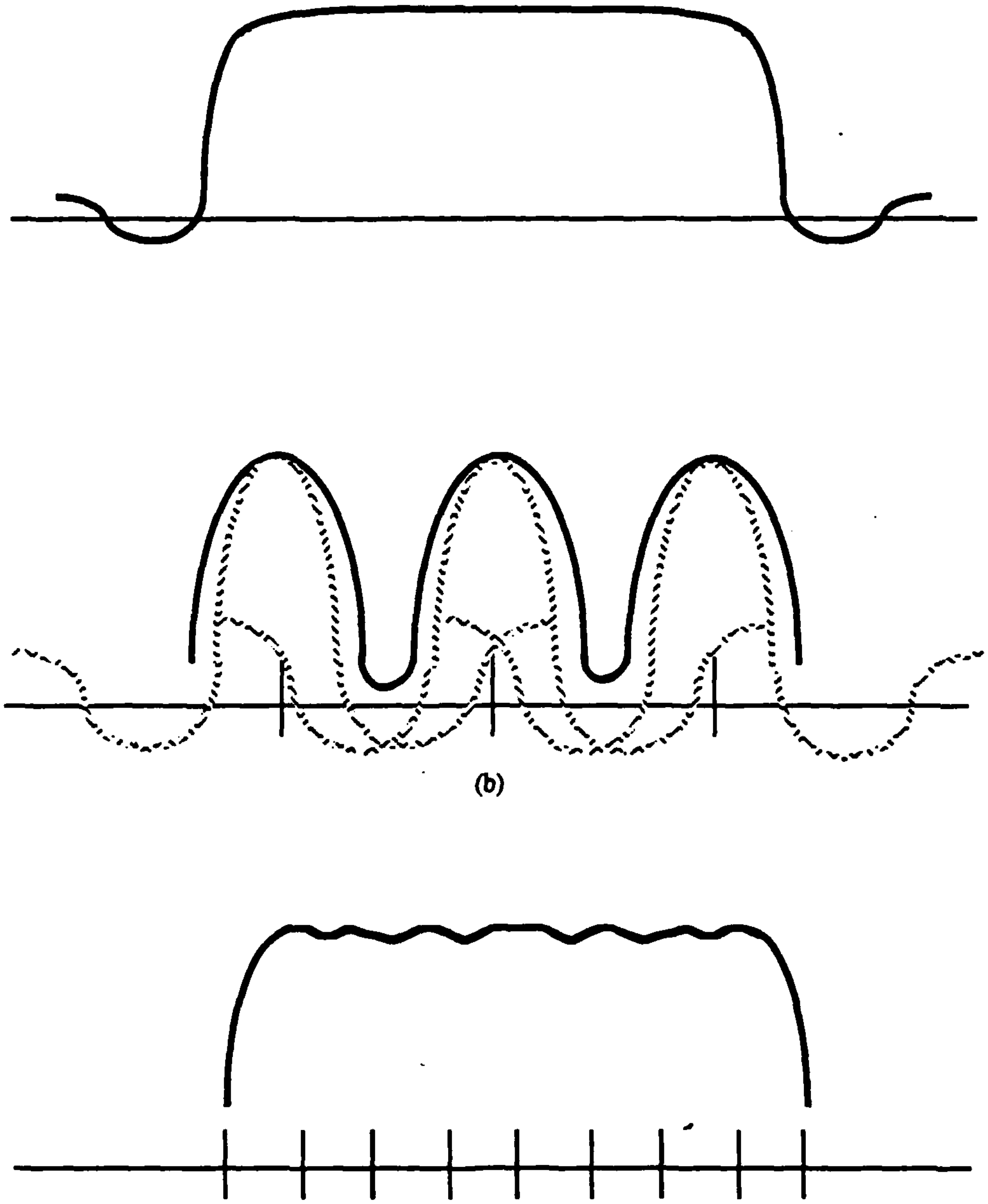
= $R \sin \theta$

Where $\theta = \cos^{-1} \left(1 - \frac{\lambda}{8R} \right)$

For a Sphere R_c = Radius of Stationary Phase Disk

For a Cylinder R_c = 1/2 of Cross-Axis Dimension of Stationary Phase Region

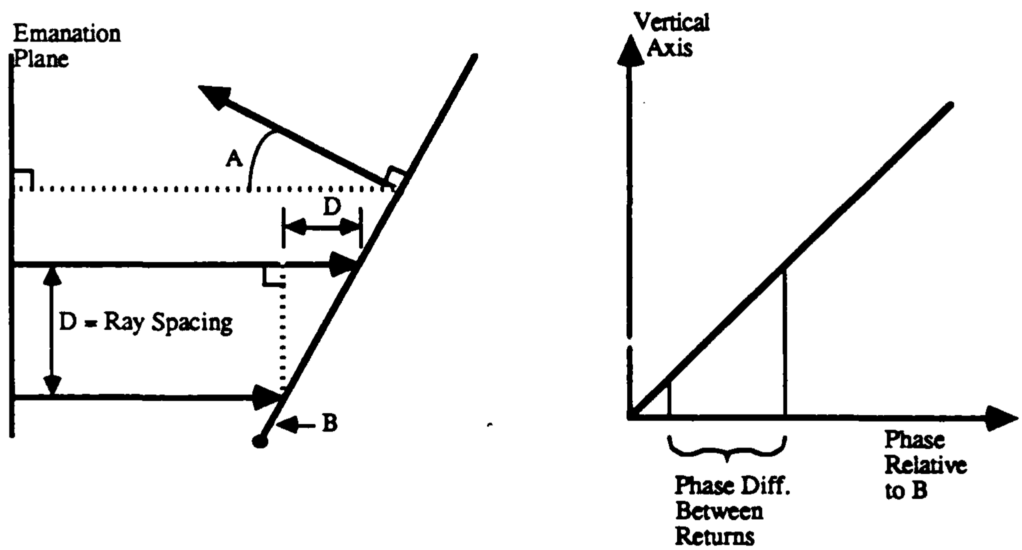
FIGURE 4-3. CHARACTERISTIC DIMENSION OF STATIONARY PHASE REGION



(b)

(c)

FIGURE 4-4.
a) PERFECTLY SAMPLED IMAGE IN AZIMUTHAL DIRECTION
b) RAY SPACING TOO LARGE (TICKS ARE SAMPLE POINTS)
c) PROPERLY SAMPLED IMAGE (TICKS ARE SAMPLE POINTS)



$D = d \tan A$

Phase Difference Between Rays $= \dfrac{2\pi}{\lambda}(2D)$

$= 4\pi d \tan A/\lambda$

Where λ = wavelength

FIGURE 4-5. PHASE RAMP FOR A FLAT SURFACE

Maximum Allowed Ray Spacing	A(in Degrees)
.142 feet	10
.069 feet	20
.043 feet	30
.030 feet	40
.021 feet	50
.014 feet	60

A = Angle Between Surface Normal and
Emanation Plane Normal.

FIGURE 4-6. MAXIMUM ALLOWED RAY SPACING FOR FLAT SURFACES AT X-BAND

happen because an undersampled set of returns will usually have phases that cause distructive interference and thus produce a good image. The radar cross section for such a case is not meaningfull except that it will be small. If the undersampling causes the returns to differ in phase by a multiple of 2*PI radians, the returns will constructively interfere and cause an artificially bright image. This is an artifact of the undersampling. An example of when this will occure is when lamda = .1 feet, ray spacing = .1 feet and A = 45 degrees. The returns will differ in phase by 4*PI radians and cause the image to appear to be an image of a plate with A = 0 degrees.

The ray spacing and pixel size can interact to cause artifacts in images. Figure 4-7 shows an example of this in the azimuth dimension. Most pixels have one return, but every 3rd pixel has 2 returns. This causes a stripped effect in the image. Curved surfaces cause the spacing in range to vary which breaks up the regularity of the pattern. Thus stripes tend to appear most often in the azimuthal direction. However, a flat surface can have this effect in both range and azimuth directions and exibit a plaid effect. These effects are usually evident in OVERLAY images. Shaded optical images are immune to these effects since one and only one ray is associated with each pixel. In RADSIM, phase cancelling and the smoothing effect of the system response convolution reduces these effects to where they are not usually visible. However, they can occure in RADSIM if the ray spacing is large enough. A good policy is to set the ray spacing at 1/5 or less of the resolution (RADSIM) or pixel size (OVERLAY). This will reduce the pixel to pixel variation in returns per pixel.

4.3 CHOOSING THE IMAGE SIZES

The image sizes for the optic subroutine of GIFT and the SHADE program are determined by the number of rays that are fired. Therefore, the size and "fineness" of the image are determined by the user responses to the "Max horiz cells" and "Max vert cells" prompts in GIFT.

In OVERLAY, the image size (in pixels) and the size of a pixel (in slant plane distance) are specified by the user (see Figures 3-15 and F-1). The product of the pixel spacing and image size gives the slant plane coverage for the image. For example, if the image is given as 128 by 128 pixels and the pixel spacing is .25 units, then the image will cover 32 by 32 distance units in the slant plane. The GIFT run that produces the ray history for an OVERLAY run will give the following information:

1. In the view plane output, GIFT gives the horizontal length of the emanation rectangle (Figure 3-9). This gives the targets extent in the azimuthal direction.
2. In the run summary information (Figure 3-10), GIFT gives the min and max ranges for 1st surfaces. This is the slant range extent for the first reflections.

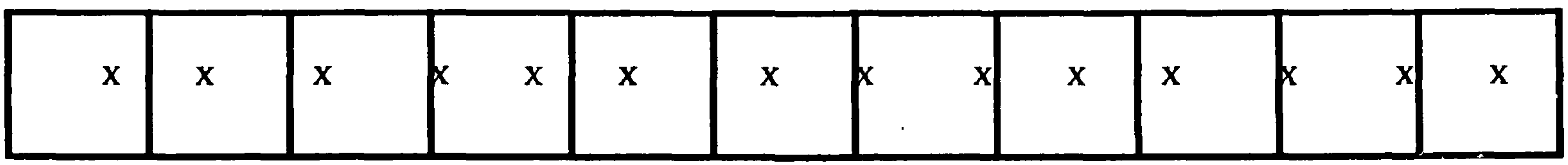


FIGURE 4-7. ROW OF PIXELS IN AZIMUTH DIRECTION
SHOWING UNEVEN RETURN COUNTS.

The image size must be large enough to cover the image. If the target is not centered in the overall coordinate system, offsets or a larger image size may have to be used. For the GIFT run of Figures 3-9 and 3-10, the target's extent in azimuth is 2 distance units while the min and max ranges for 1st reflections are .104 and .947. This gives an extent in the slant range direction of .843 units. If the pixel spacing is to be .1 unit, then an image size of 9 pixels in range and 20 pixels in azimuth will just cover the image (if the target is centered in the image). A good choice would be 32 by 32 pixels which will provide a safety factor.

In RADSIM, the azimuth extent is the horizontal length of the emanation rectangle (see discussion for OVERLAY above). However, the ranges must now include multiple reflection effects. Instead of the ranges used for OVERLAY, the GIFT output line (Figure 3-10) "min and max total ranges" is used. The max range is based on the total ray path lengths up the the last reflection. It does not include the return path to the emanation plane or the division by two that converts total range to radar range (see Sect 2..1). A multiple reflection ray is shown in Figure 2-15. The total range fund by GIFT for this ray is l1 + l2 and the slant range for the return is (l1 + l2 + l3)/2 which will always be less than the GIFT total range. Thus the max total range from GIFT gives an upper limit on the range extent. The image size can now be determined as it was for OVERLAY using min and max total range instead of min and max ranges for 1st surfaces.

An OVERLAY run be used to get a quick look at how the first reflections returns for a target will be positioned in the image. This ca be very usefull when determining the image size and offsets for RADSIM.

4.4 STANDARD FILE NAME EXTENSIONS

There are several types of files that are created when running the SRIM system of programs. In order to keep these files staight, a standard nameing convention is used. The file names are all the same and are the name of the target being run. Each file then has an extension that tells which type of file it is. These extensions are:

- CG-› Geometry description file (input to GIFT).
- 4-› Binary geometry file (produced and used by GIFT).
- RAY-› Ray history file (output by radar subroutine of GIFT).
- SLR-› Slant range optical image file (from OVERLAY).
- OPT-› Shaded optical image file (from optic subroutine of GIFT or SHADE).
- CI-› Complex image file (from RADSIM).
- MI-› Magnitude image file (from RADSIM or DETECT).
- SUR-› Surface type file (RADSIM input file).

For example, if images of an M48 tank are being run, the file names would be M48.CG, M48.4, M48.ray, M48.DI, etc.

4.5 EXAMPLE (DICYII)

A listing of DICYII.CG is shown in Figure A-3 and a view of the model is shown in Figure 4-8. This target was developed as a simple target which can be used to illustrate some the major features of SAR images using SRIM. The target is set on a plane that is used to represent the ground and shows ground clutter and shadowing. The left vertical cylinder (labelled 1) in Figure 4-8 is hollow with the bottom of the hollow at the level of the "deck" (labelled 2) that both vertical cylinders are setting on. The triangular plates are at 90 degrees to each other forming a corner reflector with the deck (labelled 3). The notch along the front is partly filled with a quarter cylinder. The ends of this cylinder along with the sides of the notch form corner reflectors (labelled 4) while the notch forms a dihedral.

4.5.1 SETTING UP THE GEOMETRY DESCRIPTION

The target consists of 9 primitives: boxes, right circular cylinders, right angle wedges, and rectangular parallelepipeds. These primitives are shown in Appendix A. Consider the notch along the front of the target. This notch could be built using any of the three methods shown in Figure 4-9. Figure 4-9a would define one region using the difference operation on two primitives, Figure 4-9b would define one region using the union operation on two primitives, and 4-9c would define two regions each containing one primitive. The two regions in Figure 4-9c are implicitly unioned by GIFT. In Section 4.1, it was pointed out that GIFT runs faster when the regions are as simple as possible. For this reason, the technique of Figure 4-9c was chosen since it does not require GIFT to explicitly evaluate any Boolean operations. In the case of the hollow cylinder, a difference operation must be used resulting in one region using two primitives.

The hollow cylinder and the deck are shown in Figure 4-10 The cylinder that is differenced to create the hollow is shown in dotted lines. Note that the base of the cylinders extend below the deck surface and that the top of the differenced cylinder extends above the top of the largest cylinder. These "quirks" are used to reduce the effects of finite precision on the ray trace done in GIFT. If the base of the cylinders were at the deck surface, a ray could get into the "crack" created by finite precision and bounce between the deck and the base of the cylinder. This would result in ray segments with zero length. The "crack" is an artifact produced by finite precision and does not really exist. When this happens the first reflection is valid, but further reflections are meaningless and are ignored by RADSIM. A similar effect can occure at the top of the hollow cylinder. If the top of the differenced cylinder was at the same height as the top of the larger

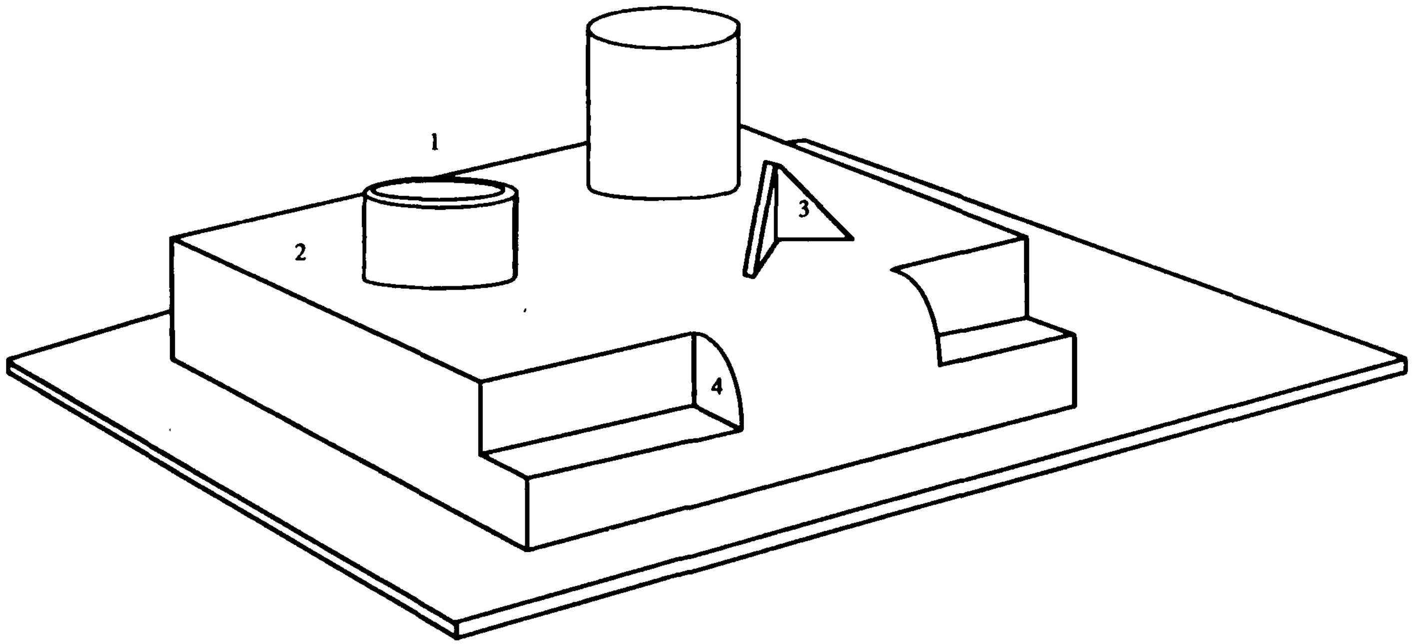


FIGURE 4-8. THE DICYII TARGET

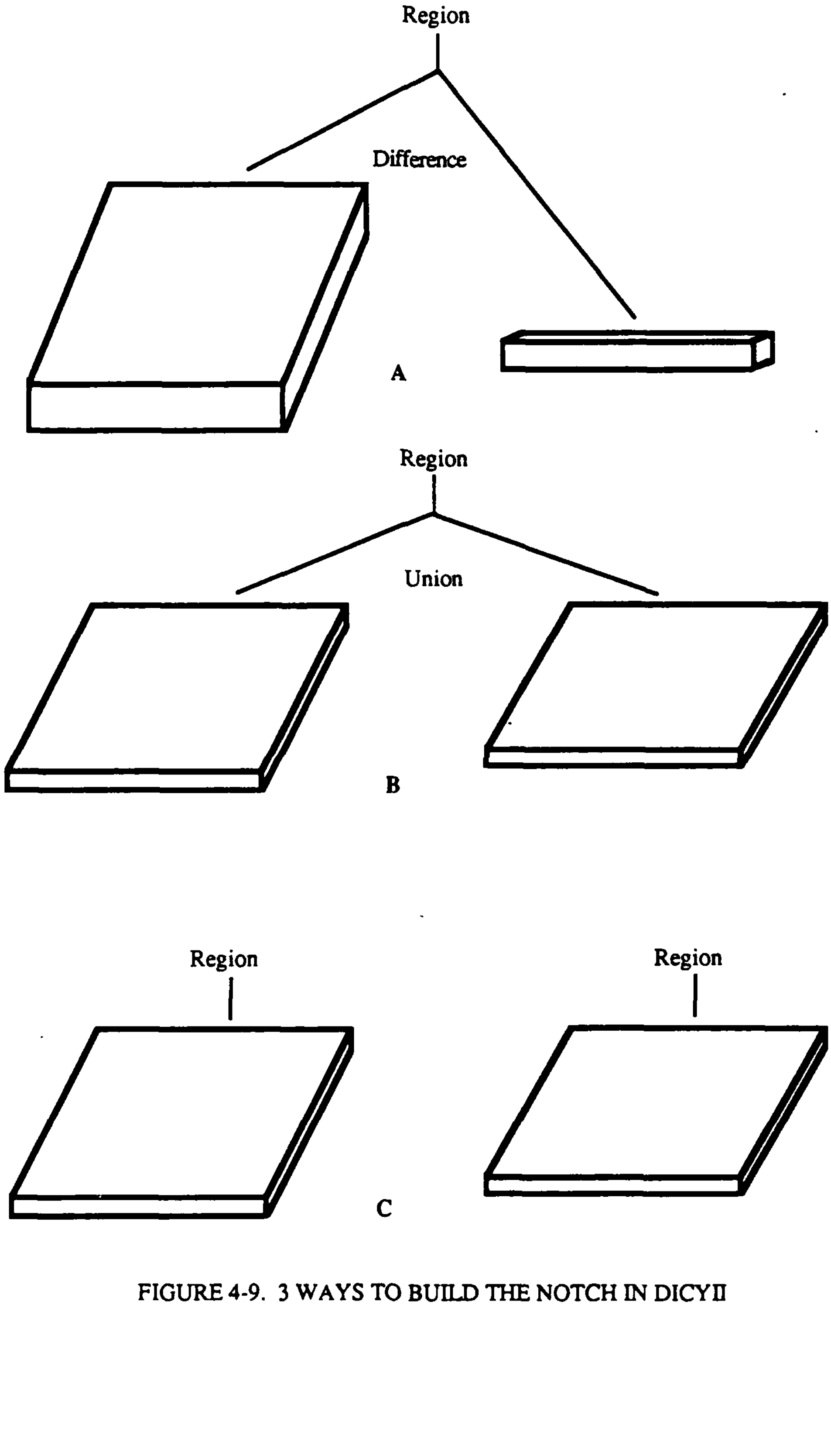
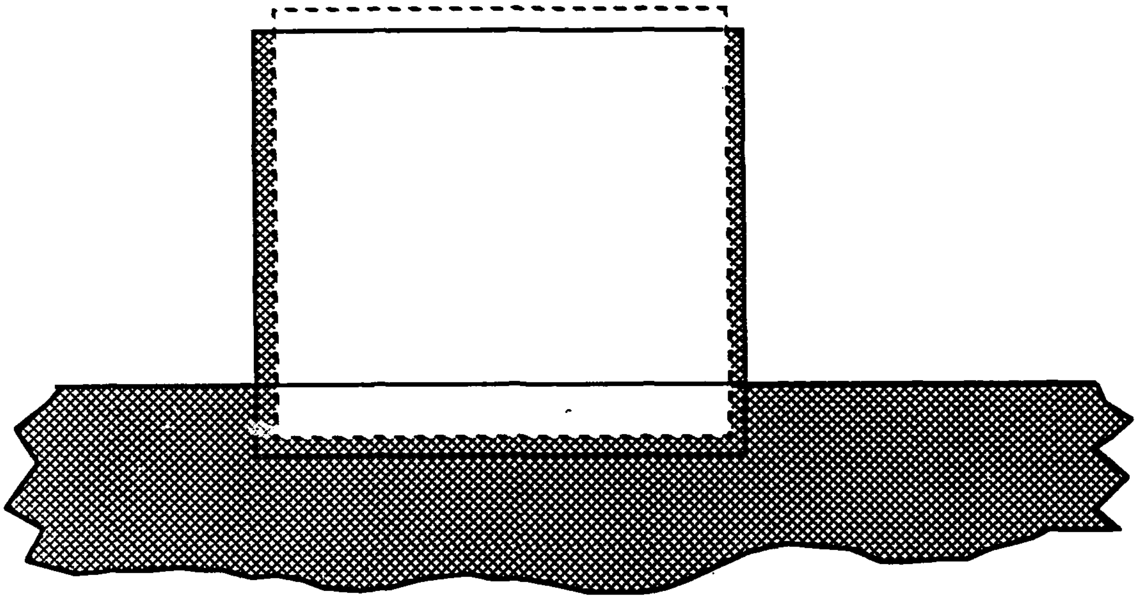


FIGURE 4-9. 3 WAYS TO BUILD THE NOTCH IN DICYII



Cross-Hatched Area is Interior of Solid

FIGURE 4-10. CROSS-SECTION OF HOLLOW CYLINDER

cylinder, finite precision could lead GIFT to think that there was an infinitesimal membrane across the opening and GIFT would reflect rays off of this membrane instead of allowing them to go into the hollow. An optical image will reveal the membrane effect while a "crack" will cause RADSIM to find "zero length ray segments" in the ray history.

These considerations lead to the geometry definition shown in Figure A-3. Note that all of the regions except one (the hollow cylinder) contain only one primitive. This model runs about twice as fast as the same model using Boolean operations to build two regions. Most models will result in some zero length ray segments. When RADSIM is run, it reports the number of these ray segments that were found in the ray history. If this number is 1% or less of the number of returns, there should not be any visible effect on the image. If the number of zero length ray segments is greater than 1%, the model should be examined to see if the number can be reduced. These zero length segment rays do not effect the optical or slant range optical images since they depend only on the first reflection which is always valid.

4.5.2 DETERMINING THE RAY SPACING

The four factors to consider for determining the ray spacing were described in Section 4.2. For optical images, the ray spacing is based on the feature size. The smallest feature in the target is the corner reflector on the deck which has right angle wedges 1.5 feet on a side. If the thickness of the wall on the hollow cylinder, the wedges, and the ground plane are ignord, a ray spacing of .5 feet will give a crude optical image showing the main features of the target. The overall dimensions of the target are 25 feet by 29 feet by 11 feet. If an image with azimuth = 0, elevation =45 is desired, the emanation plane rectangle will be about 25 feet by 21 feet (29 times sin(45) is approx. 21). This gives a ray grid of 50 by 42 for the .5 foot spacing. A ray grid of 150 by 126 will give a much finer image which should be quite good. Note that the ray grids are not symmetric. This is necessary if the scale in the image (in terms of pixels per foot) is to be the same in the horizontal and vertical directions. On ARIES, the display systems are 512 by 512. Therefore, the finest detailed image that can be produced would have a 512 by 512 grid of rays. In this case the image would not have the same scale in the horizontal and vertical directions leading to some distortion relative to a photographic image.

Suppose that an image is to be run for a x-band radar with a resolution of 5 feet. The look direction is to be (azimuth = 0, elevation = 45). The following geometric sampling consideration apply:

1. To see the corner reflector, the rays must accuratly measure the area of the triple reflection region of the reflector. The

dimensions of the corner reflector are about 1.5 feet. A ray spacing of .25 feet should be reasonable.

2. The cylinders have radii of 2 feet. For x-band, this gives the characteristic dimensions of the stationary phase region as .223 feet (from figure 4-3). To sample this requires a ray spacing of .11 feet or less.
3. If multiple reflections from one cylinder to the other are required, the spacing must be fine enough to account for the divergence introduced by the first reflection as shown in Figure 4-2. The cylinders have a radius of 2 feet and the path length between them is about 6 feet. To obtain a good image of the multiple reflection returns would require a ray spacing considerably smaller than .11 feet. considerably smaller than .11 feet. However, a ray spacing of .03 to .05 feet should show this path. The smaller ray spacing will give a better magnitude for these returns.

The above considerations result in a desired ray spacing of about .03 feet for sampling the geometry.

Sampling a system response function with an IPR of 5 feet gives a ray spacing requirement of 1 foot or less. Note that if the elevation angle is small, the ray spacing in the range direction will be spread out. For example, if the elevation was 20 degrees then a vertical ray spacing in the emanation plane of 1 foot would lead to a ray spacing in the slant range plane range coord. of (1 foot)/tan(20) = 2.75 feet on the deck and the ray spacing for GIFT would have to be decreased to account for this. The case of elevation = 45 degrees gives a range spacing equal to the emanation plane spacing. If the image will have two pixels per IPR, then the pixel size will be 2.5 feet and a ray spacing of .5 feet should also give a smooth variation in return count per pixel.

The phase sampling along the flat surfaces is the the last item to be considered. All of the flat surfaces have normals that make an angle of 45 degrees with the emanation plane normal. The table in Figure 4-6 gives a ray spacing requirement of .025 or less.

Now combining the results from the above considerations, the minimum ray spacing is established by the phase sampling and is less than .025 feet. Therefore, the ray grid should be greater than 1000 by 840 rays (for an emanation rectangle of 25 feet by 21 feet). If this is too many rays, two things can be done. The first is to note that the phase sampling depends only on the vertical ray spacing since only rays with different vertical coordinates in the emanation plane will cause returns with different ranges on the flat surfaces in the slant range plane. Thus the horizontal ray spacing can be increased to the next smallest requirement which was the geometry sampling requirement. If this is taken to be .04, then the ray grid can be reduced to being 700 by 840. The second is to allow

the phase to be undersampled and count on this not effecting the image. The effects of these two types of undersampling are treated in the next paragraph

The two criteria that lead to low ray spacing are the geometry sampling for multiple reflections from curved surfaces and the phase sampling on flat surfaces. If the geometry is undersampled, the radar cross section becomes less accurate and weak returns (due to large divergence factors between reflections) may drop out of the image entirely. However, the image quality degrades gracefully with geometry undersampling and will still produce qualitatively good images for significant undersampling. Undersampling the phase on flat surfaces does not usually harm the image. However, it introduces the possibility of a catastrophic failure due to the interaction of the wavelength and ray spacing on the flat surface as described in Section 4.2. The probability of this happening is reduced if the ray spacing and wavelength are not even multiples of each other. Therefore, if the wavelength is .1 foot and the ray spacing is .1 foot, one or the other number should be altered by a few percent.

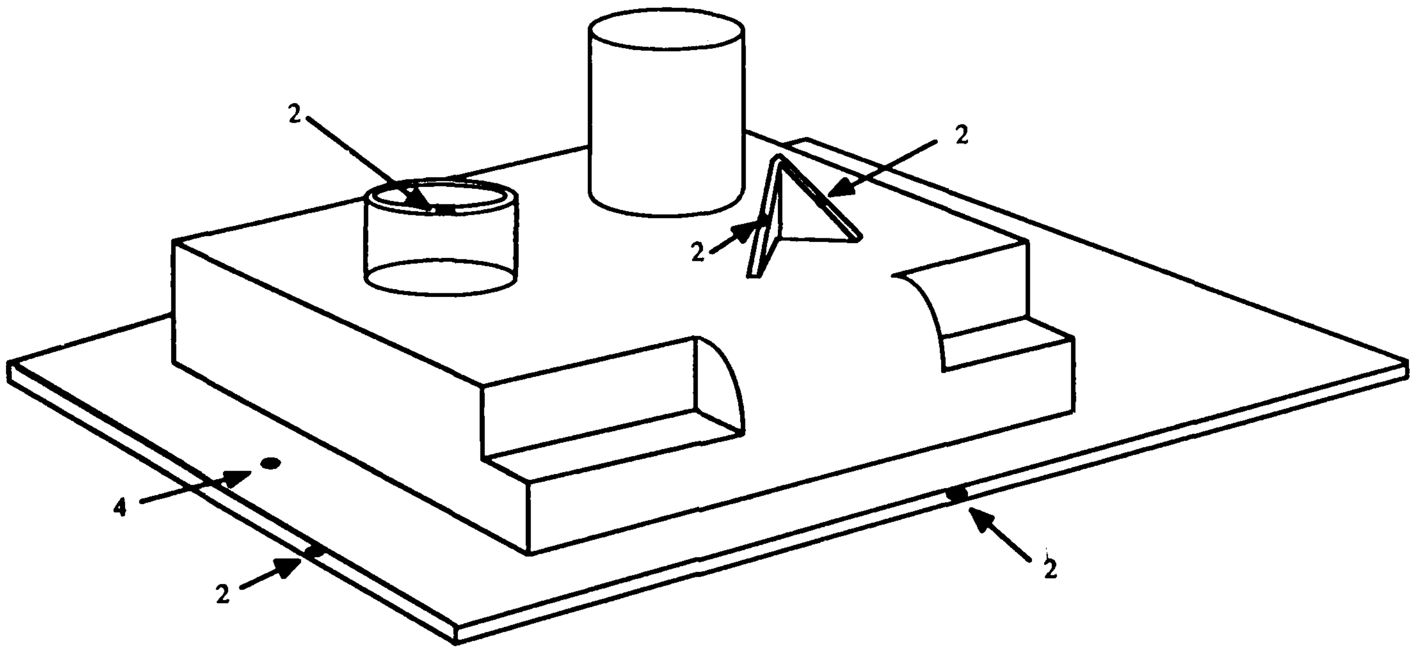
In practice, good images showing the multiple reflection return from the cylinders and giving qualitatively correct phase cancelling have been produced on this target by firing 512 by 512 grids of rays. This illustrates both the high number of rays required to produce rigorously correct ray sampling on a relatively simple target and that one can usually "get away" with violating the sampling requirement to some extent. The number of rays fired is often determined by how many rays the user can afford to fire given the program run times and the size of the ray history file.

4.5.3 SETTING UP THE SURFACE TYPE FILE

The purpose of the surface type file is to assign a model from the reflection model file to the surfaces in the target. These files are described in Appendix G. If the surface type file does not contain a reflection model assignment for a surface, the surface defaults to a "smooth reflector" in RADSIM. If all of the surfaces are smooth reflectors, a surface type file is not required.

For DICYII, the thin edges of the hollow cylinder, right angle wedges, and the ground plane are set to perfectly absorbing (reflection model 2) while the top surface of the ground plane is set to a rough surface model (reflection model 3). Note that modeling these edges would require diffraction which RADSIM does not handle. Since the electromagnetic scattering of these edges cannot be modeled, it is better to eliminate them entirely by making them perfectly absorbing. Figure 4-11 shows a view of DICYII with the absorbing and rough surfaces labeled. All other surfaces are smooth reflectors.

By using the information from Appendix A, the surface



2 = Perfect Absorber
4 = Rough Surface (ground clutter)

All other surfaces are smooth reflectors

FIGURE 4-11. REFLECTION MODEL ASSIGNMENTS FOR DICYII

numbers for these surfaces can be identified and the surface type file shown in Figure G-1 created. The surfaces assigned to the smooth reflector model do not need to be in the file (RADSIM will default them to smooth reflectors), but are included to accomodate possible changes in their reflection model assignments.

APPENDIX A
GEOMETRY DESCRIPTION FILE CONTENTS

This section contains the information available at ERIM on the primitives (covered in Section A.1) and on the geometry description file (covered in Section A.2).

A.1 PRIMITIVE SOLIDS

The primitives that can be used to build targets are shown in Figure A-1. The primitives are specified by giving values to the variables shown in the Figure. The assignment of surface numbers is shown in Figure A-2. These are the surface numbers used when assigning reflection models to the surfaces in the surface type file used by RADSIM.

A.2 GEOMETRY DESCRIPTION FILE

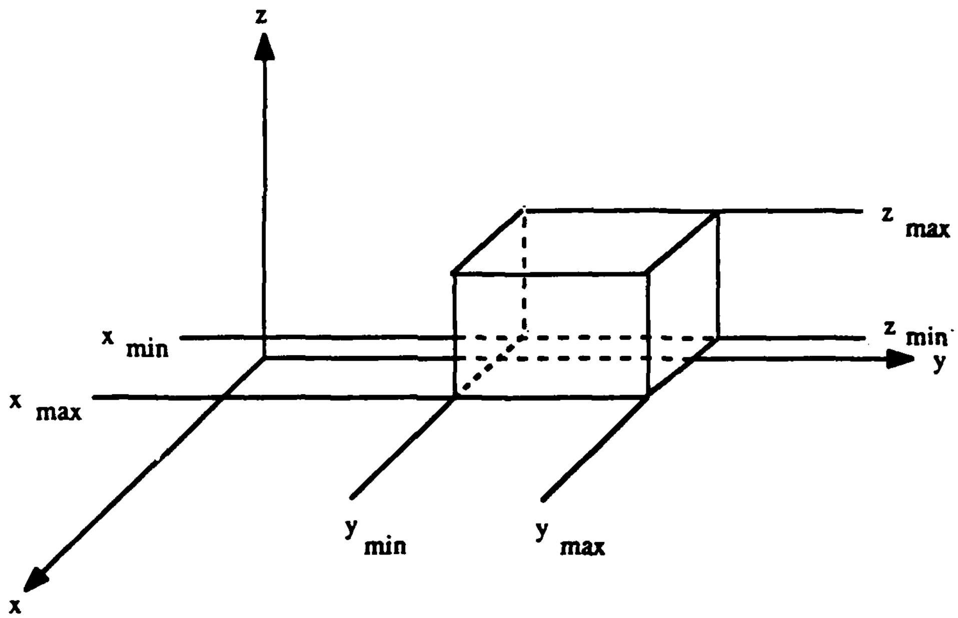
The geometry description file contains five record types. The geometry description file for the DICYII target is shown in Figure A-3. The labels to the right of the Figure indicate the record types.

The first record is the TITLE record and is in the format shown in Figure A-4. If the units are always set to "m", no units conversion is done in GIFT. Thus the units are always set to "m" regardless of the actual units. The actual units are indicated in the "name for target" field as shown in Figure A-3. The second record is the CONTROL record and has the format shown in Figure A-4. This record contains the number of primitives and regions used to build the target.

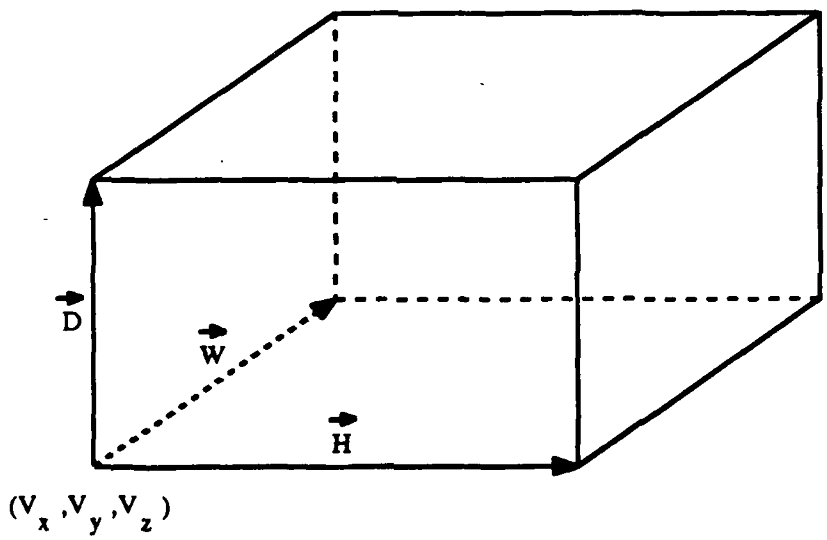
The next records contain the primitives to be used. The record format is shown in Figure A-4 and specification for each primitive is shown in Figure A-5. The symbols used in this Figure are defined in Figure A-4 and are used in Figure A-1.

The next records specify how the primitives are combined to form the regions. If a region definition requires more than one record, the continuation records will not have an entry for the "region number" field. A record with a "-1" in columns 4-5 signals the end of the region definitions.

The la' records are region identification records. There is one record for each region. Currently, the component code number used is always 501 and all other items are zero or blank except for the region description as shown in Figure A-3.

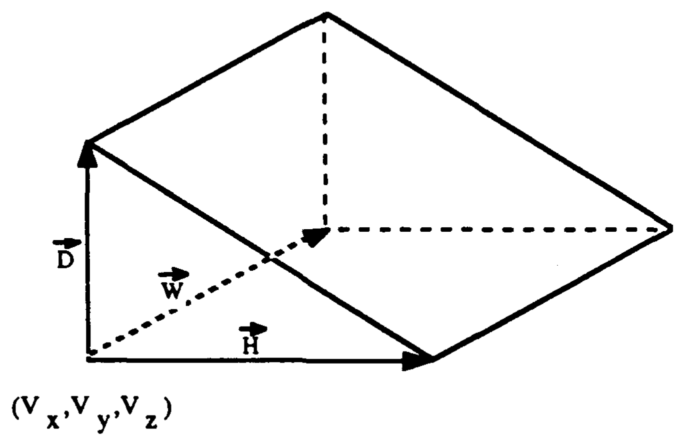


Rectangular Parallelepiped (RPP)

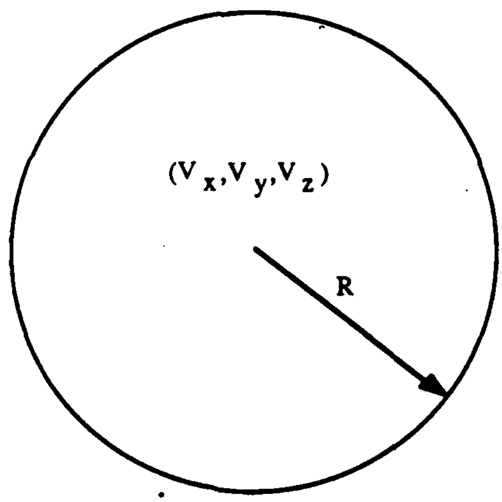


Box

FIGURE A-1. PRIMITIVE SOLIDS.

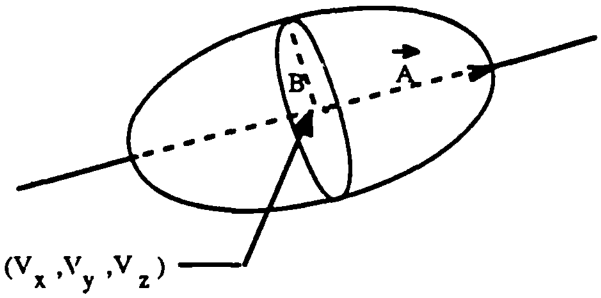


Right Angle Wedge (RAW)

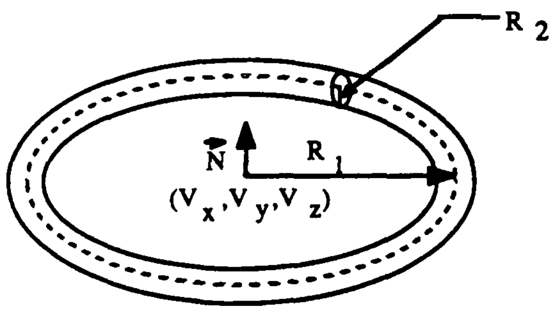


Sphere (SPH)

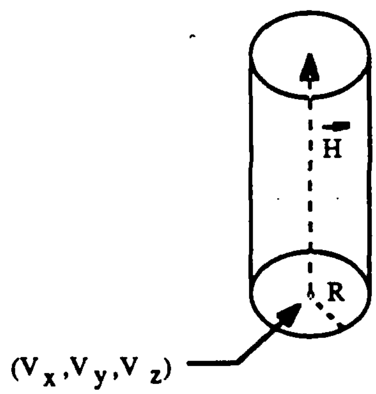
FIGURE A-1. (CONTINUED)



Ellipse (ELL)

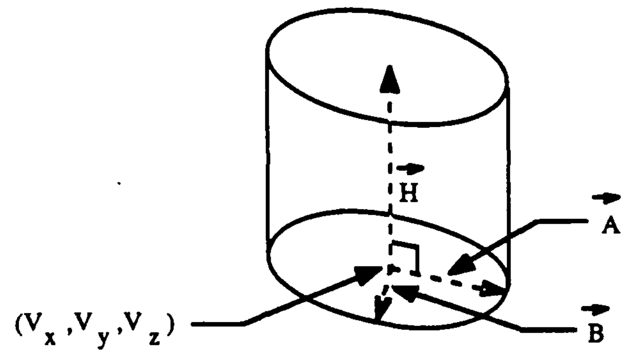


Torus (TOR)

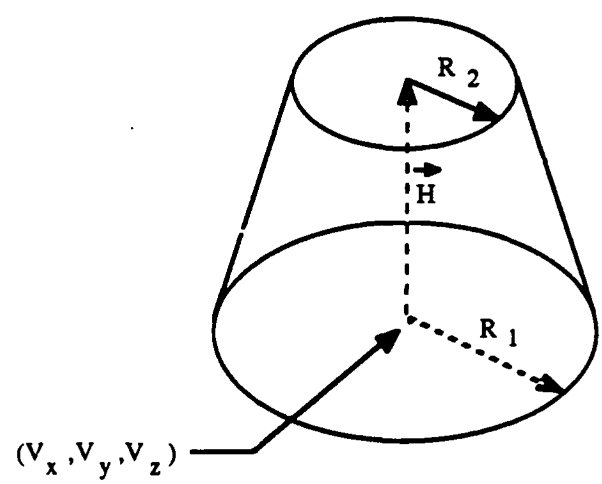


Right Circular Cylinder (RCC)

FIGURE A-1. (CONTINUED)



Right Elliptical Cylinder (REC)



Truncated Right Cone (TRC)

FIGURE A-1. (CONTINUED)

$\hat{N} = A\hat{x} + B\hat{y} + C\hat{z}$ (Unit Vector)
$\hat{x}, \hat{y}, \hat{z}$ = Coordinate System Unit Vectors

D = Distance from Center of
Coordinate System to Plane
Along a Normal to the Plane

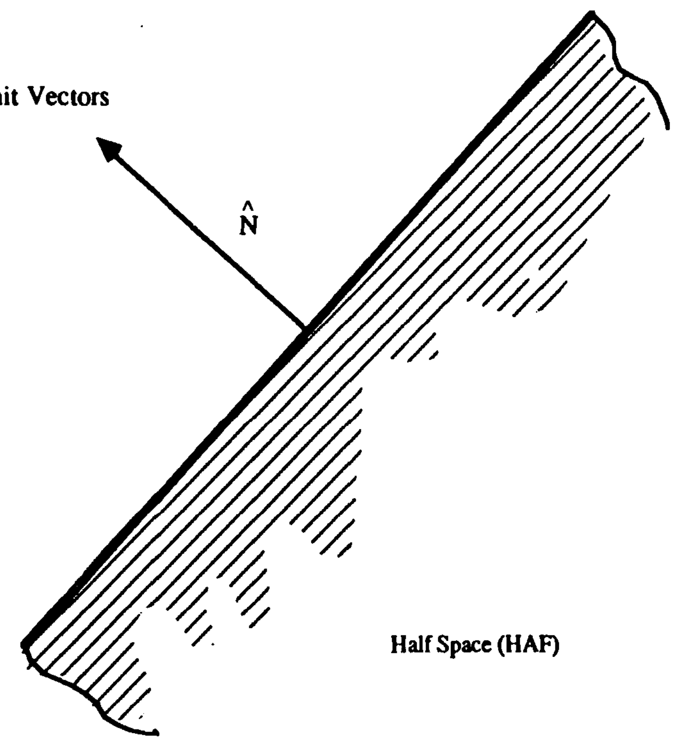


FIGURE A -1. (CONTINUED)

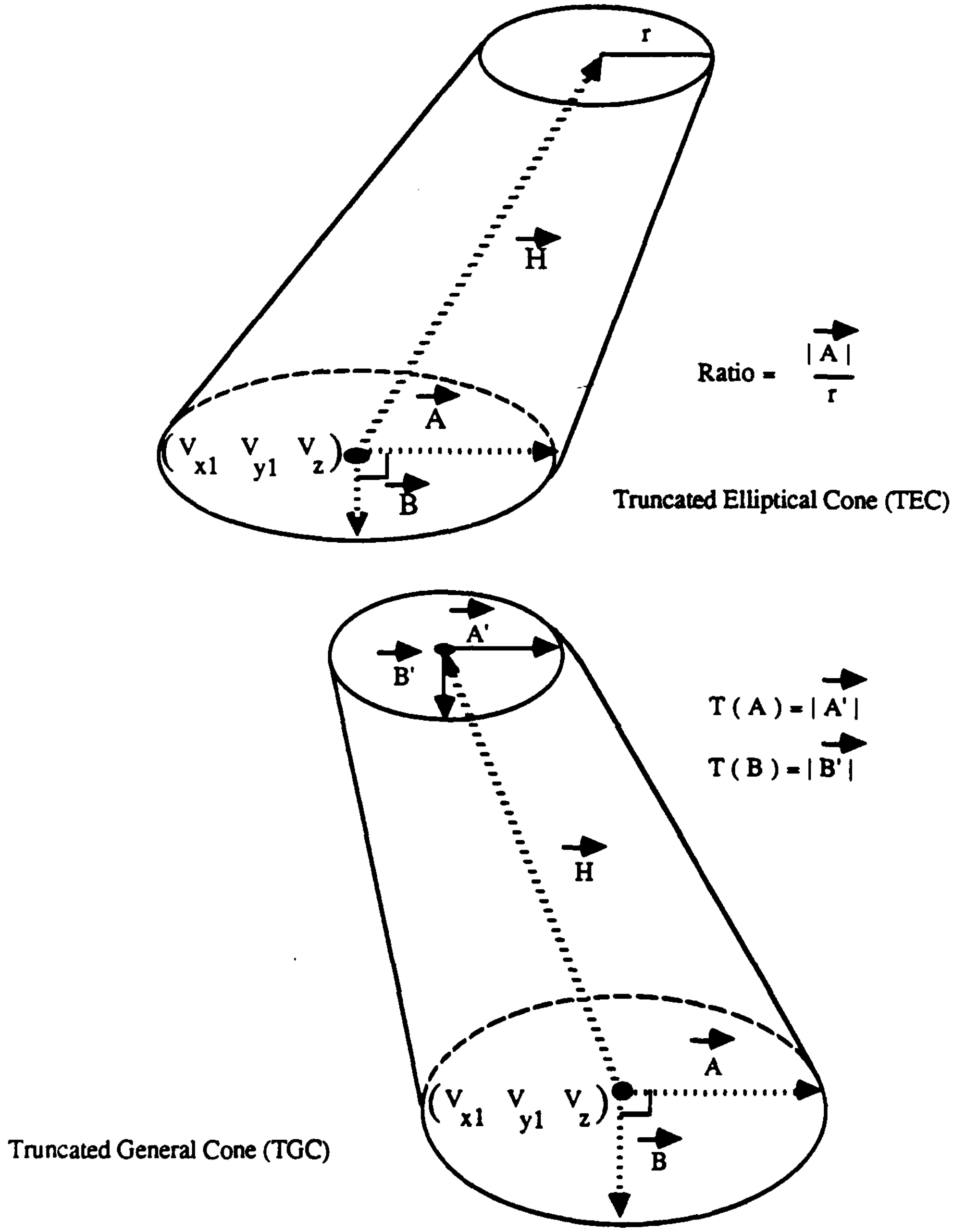
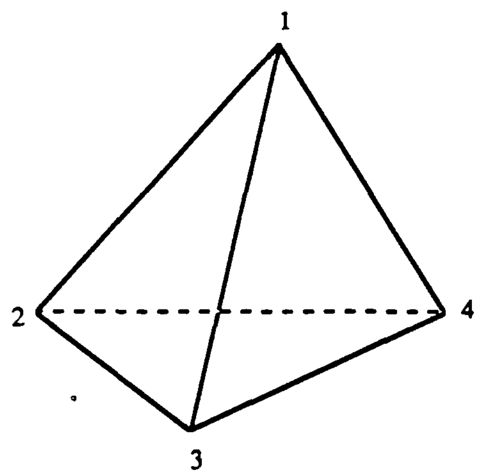
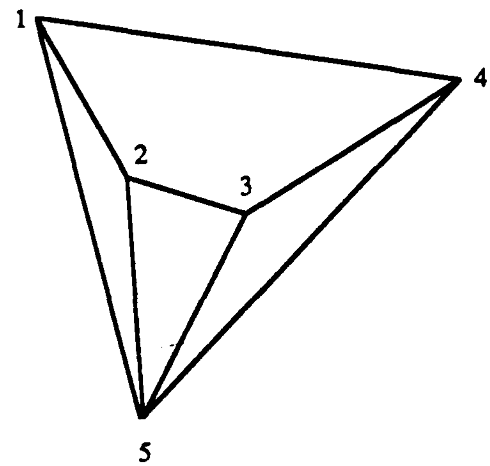


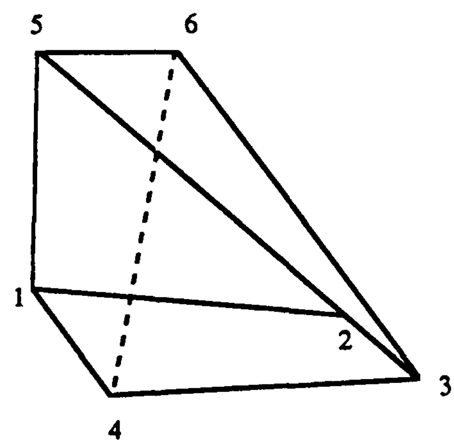
FIGURE A-1. (CONTINUED)



ARB 4

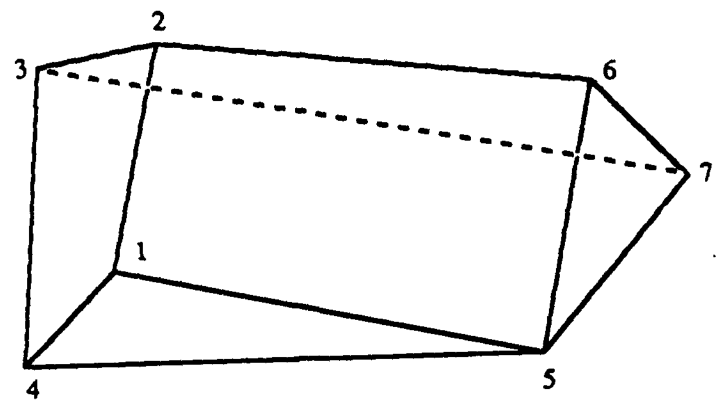


ARB 5

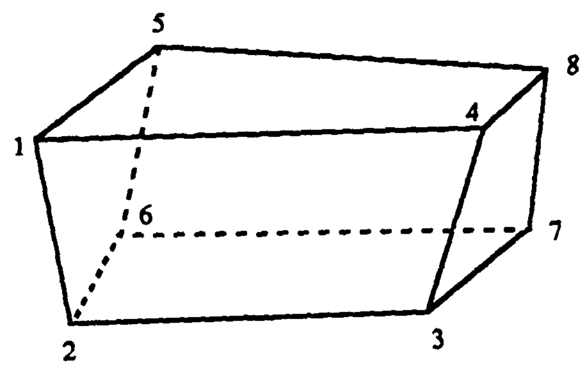


ARB 6

FIGURE A-1. (CONTINUED)



ARB 7



ARB 8

FIGURE A-1. (CONTINUED)

BODY \ SURF. #	1	2	3	4	5	6
RRP	Plane $x = x\ min$	Plane $x = x\ max$	Plane $y = y\ min$	Plane $y = y\ max$	Plane $z = z\ min$	Plane $z = z\ max$
BOX	Plane of $\vec{H}, \vec{D}, V$	Plane of $\vec{H}, \vec{D}$, not V	Plane of $\vec{W}, \vec{D}, V$	Plane of $\vec{W}, \vec{D}$, not V	Plane of $\vec{H}, \vec{W}, V$	Plane of $\vec{H}, \vec{W}$, not V
RAW	Plane of $\vec{D}, \vec{W}, V$	Plane of $(\vec{H}-\vec{D}), \vec{W}$ not V	Plane of $\vec{H}, \vec{W}, V$		Plane of $\vec{D}, \vec{H}, V$	Plane of $\vec{D}, \vec{H}$, not V
SPH	Spherical Surface					
ELL	Ellipsoidal Surface					
TOR	Torus Surface					
RCC	Plane Containing V	Plane Not Containing V	Cylindrical Surface			
REC	Plane Containing V	Plane Not Containing V	Cylindrical Surface			
TRC	Plane Containing V	Plane Not Containing V	Conic Surface			
TEC	Plane Containing V	Plane Not Containing V	Curved Surface			
TGC	Plane Containing V	Plane Not Containing V	Curved Surface			
HAF	Half Plane Surface					

FIGURE A-2. SURFACE NUMBERS FOR PRIMITIVES (CONTINUED)

BODY \ SURF. #	1	2	3	4	5	6
ARB 4	123	412	423	431		
ARB 5	FACE 1234	FACE 512	FACE 523	FACE 524	FACE 541	
ARB 6	FACE 1234	FACE 2365	FACE 1564	FACE 512	FACE 634	
ARB 7	FACE 1234	FACE 567	FACE 145	FACE 2376	FACE 1265	FACE 4375
ARB 8	FACE 1234	FACE 5678	FACE 1584	FACE 2376	FACE 1265	FACE 4378

FIGURE A-2. SURFACE NUMBERS FOR PRIMITIVES (CONTINUED)

```
m    ft newfangled dicy test object 11-13-84        TITLE
     9    8                                          CONTROL
     1box        -9.0,-8.0,-5.1,18.0,0.0,0.0
     1            0.0,15.0,0.0,0.0,0.0,3.1
     2box        -9.0,-6.0,-2.1,18.0,0.0,0.0
     2            0.0,14.0,0.0,0.0,0.0,2.1
     3rcc        -1.0,-5.0,-2.0,6.0,0.0,0.0
     3            2.0
     4rcc        -4.0,3.0,-.1,0.0,0.0,3.1
     4            2.0                                PRIMITIVES
     5rcc        -4.0,3.0,0.01,0.0,0.0,3.1
     5            1.9
     6rcc         4.0,3.0,-.1,0.0,0.0,3.1
     6            2.0
     7raw         2.5,-1.0,-.1,0.0,0.0,2.1
     7           1.49,-1.49,0.0,.007,.007,0.0
     8raw         2.5,-1.0,-.1,0.0,0.0,2.1
     8          -1.49,-1.49,0.0,-.007,.007,0.0
     9rpp       -13.5,13.5,-10.6,14.5,-5.01,-5.0
     1        1
     2        2
     3        3
     4        4     -5                               REGIONS DEFINITIONS
     5        6
     6        7
     7        8
     8        9
    -1
     1   501    0                object
     2   501    0                object
     3   501    0                object              REGION IDENTIFICATION
     4   501    0                object
     5   501    0                object
     6   501    0                object
     7   501    0                object
     8   502    0                object
```

FIGURE A-3. THE DICYII GEOMETRY DESCRIPTION FILE.

TITLE record (A2,3X,A60)

cols.	contents
1-2	Target units
6-65	Name for target

CONTROL record (2I5)

cols.	contents
1-5	Number of primitives
6-10	Number of regions

PRIMITIVE records (A5,A3,A2,6F10.0,A10)

REGION definition records (I5,1X,9(A2,I5),1X,A10)

cols.	contents
1-5	Region number
7-8	Operator
9-13	Primitive or region number
14-15	Operator
16-20	Primitive or region number
21-22	Operator
23-27	Primitive or region number
28-29	Operator
30-34	Primitive or region number
35-36	Operator
37-41	Primitive or region number
42-43	Operator
44-48	Primitive or region number
49-50	Operator
51-55	Primitive or region number
56-57	Operator
58-62	Primitive or region number
63-64	Operator
65-69	Primitive or region number
71-80	Comments

Operators are:

"-" -> Difference
"or" -> Union
"+" -> Intersection

Figure A-4
Geometry Description Record Formats

REGION IDENTIFICATION record (5I5,5X,A40)

cols.	contents
1-5	Region number
6-10	Component code number (1-9999)
11-15	Space code number (1-99)
16-20	Material code
21-25	Percent
31-70	Description of region

Figure A-4 (continued)
Geometry description file record formats

symbols used in this table:

- V -> Vertex
- H -> Height vector
- W -> Width vector
- D -> Depth vector
- N -> Normal vector
- A -> Semi-major axis of ellipse
- B -> Semi-minor axis of ellipse
- R -> Radius
- T -> Length of top axis with same direction as bottom axis

cols 1-5	6-8	9-10	11-20	21-30	31-40	41-50	51-60	61-70	71-80
No.	RPP		xmin	xmax	ymin	ymax	zmin	zmax	comments
No.	BOX		Vx	Vy	Vz	Hx	Hy	Hz	comments
No.			Wx	Wy	Wz	Dx	Dy	Dz	comments
No.	RAW		Vx	Vy	Vz	Dx	Dy	Dz	comments
No.			Hx	Hy	Hz	Wx	Wy	Wz	comments
No.	SPH		Vx	Vy	Vz	R			comments
No.	ELL		Vx	Vy	Vz	Ax	Ay	Az	comments
No.			R						comments
No.	TOR		Vx	Vy	Vz	Nx	Ny	Nz	comments
No.			R1	R2					comments
No.	RCC		Vx	Vy	Vz	Hx	Hy	Hz	comments
No.			R						comments
No.	REC		Vx	Vy	Vz	Hx	Hy	Hz	comments
No.			Ax	Ay	Az	Bx	By	Bz	comments
No.	TRC		Vx	Vy	Vz	Hx	Hy	Hz	comments
No.			R1	R2					comments
No.	TEC		Vx	Vy	Vz	Hx	Hy	Hz	comments
No.			Ax	Ay	Az	Bx	By	Bz	comments
No.			RATIO						comments
No.	TGC		Vx	Vy	Vz	Hx	Hy	Hz	comments
No.			Ax	Ay	Az	Bx	By	Bz	comments
No.			T(A)	T(B)					comments
No.	HAF		A	B	C	D			comments

equation for plane is $Ax + By + Cz = D$

Figure A-5
Primitive Solid Definitions

cols 1-5	6-8	9-10	11-20	21-30	31-40	41-50	51-60	61-70	71-80
No.	ARB	4	X1	Y1	Z1	X2	Y2	Z2	comments
No.			X3	Y3	Z3	X4	Y4	Z4	comments
No.	ARB	5	X1	Y1	Z1	X2	Y2	Z2	comments
No.			X3	Y3	Z3	X4	Y4	Z4	comments
No.			X5	Y5	Z5				comments
No.	ARB	6	X1	Y1	Z1	X2	Y2	Z2	comments
No.			X3	Y3	Z3	X4	Y4	Z4	comments
No.			X5	Y5	Z5	X6	Y6	Z6	comments
No.	ARB	7	X1	Y1	Z1	X2	Y2	Z2	comments
No.			X3	Y3	Z3	X4	Y4	Z4	comments
No.			X5	Y5	Z5	X6	Y6	Z6	comments
No.			X7	Y7	Z7				comments
No.	ARB	8	X1	Y1	Z1	X2	Y2	Z2	comments
No.			X3	Y3	Z3	X4	Y4	Z4	comments
No.			X5	Y5	Z5	X6	Y6	Z6	comments
No.			X6	Y6	Z6	X7	Y7	Z7	comments

Figure A-5
Primitive solid definitions (continued)

APPENDIX B
RAY HISTORY FILE CONTENTS

The general content of a ray history file is a history of information for each ray that is fired. Each ray history consists of a sequence of logical records including a firing point record, possibly some reflection records, and possibly an escape record. Since a ray history may be terminated either by an escape or by reaching the maximum allowed reflections, some ray histories have no escape record. A ray history is not output for a ray that has no reflections. In general, a typical ray history will include each type of logical record. Each view contains three header records, a set of ray histories, an end of view record, and an end of file marker. A ray history file may contain more than one view. A ray history file is:

‹ray file› = ‹view 1›‹view 2› ...
‹ray view› = ‹headers›‹ray history›‹ray history› ...
‹ray history›‹end of view›‹end of file marker›
‹ray history› = ‹firing›‹reflection›‹reflection› ...
‹escape›

Each logical record contains 18 words of data. The contents of the logical records in a ray history are shown in Figure B-1 through B-5

Header record number 1 - View data

word	contents
0	The ascii string "head"
1	Run ID (GIFT user input)
2	View ID (GIFT user input)
3-5	Coordinates of the target center
6	Back off distance
7	Emanation plane elevation angle
8	Emanation plane azimuth angle
9-10	Emanation rectangle horizontal and vertical dimensions
11-12	Number of ray cells in horizontal and vertical directions.
13	Maximum allowed number of reflections
14	View number (set by GIFT)
15-17	Zeros

Header record number 2 - Ascii string (user specified comment)

Header record number 3 - Date and time of the GIFT run

Figure B-1
Contents of the header records

word	Contents
0	The ascii string "fire"
1	Zero
2	Number of logical records following the firing record. Usuall equals number of reflections plus one but can be number reflections if no escape record is present.
3-5	Coordinates of the firing point in the overall coordinate system.
6-7	Coordinates of the firing point in the emanation plane coordinate system.
8	Column number of ray in ray grid.
9	Row number of ray in ray grid.
10-17	Zeros

Figure B-2
Contents of the ray firing records

word	contents
0	The ascii string "refl"
1	The body type. This is the index into the GIFT body table.
2	Line of sight flag. This is 1 if an unobstructed line of sight exists from the reflection point back to the emanation plane along the emanation plane normal.
3-5	Coordinates of the reflection point in the overall coordinate system.
6-8	Components of the surface unit normal at the reflection point.
9-11	Unit vector in direction of the target surfaces major axis of curvature.
12-13	Major and minor Gaussian curvatures for the target surface at the reflection point.
14	Ray path length from the previous reflection or from the emanation plane.
15	Region number for the region of the target containing the reflection point.
16	Body number for the primitive solid that the ray reflected from.
17	The surface number of the primitive solid's surface that the ray reflected from.

Figure B-3
Contents of the *reflection* records

word	contents
0	The ascii string "escp"
1	The escape flag -1 -> Ray crossed the emanation rectangle -2 -> Ray crossed emanation plane outside of the emanation rectangle. -3 -> Ray did not cross the emanation plane.
2-8	Zeros
9-11	Unit vector in direction of ray after the last reflection.
12-17	Zeros.

Figure B-4
Contents of escape records

End of physical record	- A logical record containing a -1 in word 17 flags the end of the current physical record. The contents of words 0 through 16 should be ignored.
End of view record	- This record has the ascii string "endv" in word zero. The contents of words 1 through 16 should be ignored.

Figure B-5
Contents of end-physical record
and end-of-view logical records

APPENDIX C
GIFT WORK SESSION

This Appendix contains annotated work sessions for GIFT. The first work session shown in Figure C-1 is for an run to generate a shaded optical image for the DICYII target. The second work session shown in Figure C-2 is to create a ray history file for the DICYII model to be used by the other SRIM programs. The user inputs are underlined and the circled numbers in the Figures correspond to the notes that are given below.

Notes for Figure C-1:

1. This is the run command to the VMS operating system.
2. The user enters either the geometry definition file name of the binary file name. In this case the user has entered the geometry definition file name.
3. This is the GIFT startup message.
4. This is the option prompt for debug output and tolerance setting. The response indicates no options are to used.
5. Output from the "geni" subroutine while reading in the geometry definition file. If debug options had been selected in item 4, additional debug output would be written here.
6. Prompt sequence and response for selecting the GIFT function. GIFT lists the subroutine option and then asks which one the user wants. In this case the "optic" option has been selected (produce shaded optical image file).
7. This is the optic routine start up message.
8. The user enters the optical image file name. The optical image file will be a displayable magnitude file of 16 bit words.
9. This is the azimuth angle (Figure 2-6) to be used for the ray trace. This along with the elevation angle defines the emanation plane normal.
10. This is the elevation angle (Figure 2-6) to be used for the ray trace. This along with the azimuth angle defines the emanation plane normal.
11. The number of rays to fire in the horizontal direction.
12. The number of rays to fire in the vertical direction.
13. The azimuth angle (Figure 2-6) for the illumination direction. This defines a vector to the illuminator.
14. The elevation angle (Figure 2-6) for the illumination direction. This defines a vector to the illuminator.
15. This is the output from optic as it determines the emanation plane, the emanation rectangle, the illumination direction, and the ray spacing.
16. Prompt to confirm setup. In this case the setup is OK and the user enters "c" to continue with the shaded image generation.

```
run [grim.gift]gift                              1
Enter geometry file name: ? dicyii.cg                     2

GIFT Program - VAX/VMS FORTRAN 77 Version
Program set  1985 February 12                           3
Execution date -  12-APR-85


Input for Main
s - Print solids        r - Print Regions
i - Print Idents        a - Print Aster array
o - Print ordered id    e - Print ordered reg rpps         4
t - Overlap tol         l - LOS Tolerance
m - Max errors (25)
Main Option (End=0)? 0

Enter geni

Title - ft newfangled dicy test object 11-15-84
Target units (in)

Number of solids     9
Number of regions    8


** Diagnostic ** The following regions are null                5


Total storage for geometry data                 248
Total working storage                            35
Total storage in master-aster                   275

Tolerance for overlap          tol    =        0.0100
Tolerance for line of sight    tollos =        0.0100

Processed data written on file dicyii.4
Time for input processing     0.00 seconds
Leave geni

Application Subroutines
end      brandx   check   optic   radar                  6
Application Subroutine? optic

Enter optical

optical version 27-feb-1985
modified to keep last ray                                      7
print range information
see George Darling about any problems
Bring hardcopy of results
```

FIGURE C-1. EXAMPLE GIFT (OPTIC) RUN.

```
Enter optical image file name: (<retn> = default) dicyii.mi   8
Azimuth (degrees)? 30.     9
Elevation (degrees)? 45.   10
Max horz cells? 128   11
Max vert cells? 128   12
illumination azimuth (degrees)? 0.   13
illumination elevation (degrees)? 90.   14

Azimuth  30.0
Elevation  45.0

Target parameters          x          y          z
  Minimum              -14.500    -13.500     -5.100
  Maximum               10.600     15.500      6.000
  Center                -1.950      1.000      0.450
  Dimensions            25.100     29.000     11.100

View plane
  Horizontal length     37.665
  Vertical length       33.472
  Depth                 33.472
  Back off distance     17.887
  Center                 1.841      1.159                15
  Horz Cell size         0.294
  Vert Cell size         0.262
  Horizontal range     -16.991     20.673
  Vertical range       -15.577     17.895

Horz number cells          128
Number vert cells          128

Number of cells          16384
Light source
  white                    255
  gray                     128
  black               0
  source direction     0.000      0.000      1.000
Do you r(estart,c(ontinue or e(nd: ? c    16

Time for optical     0.00 seconds
                                                         17
Leave optical

Application Subroutines
end     brandx  check   optic   radar                    18
Application Subroutine? end

End of run
```

FIGURE C-1. EXAMPLE GIFT (OPTIC) RUN.
(Concluded)

17. Ending message from the optical subroutine.
18. Prompt sequence for subroutine selection. The user is done and tells GIFT to terminate execution.

Notes for Figure C-2:

1. See Figure C-1 note.
2. See Figure C-1 note.
3. See Figure C-1 note.
4. See Figure C-1 note.
5. See Figure C-1 note.
6. Prompt sequence for selecting the GIFT subroutine to be executed. In this case the radar subroutine has been selected to produce a ray history file.
7. Radar subroutine startup message.
8. Number of views to be run. This should always be one for production work.
9. Maximum number of reflections to be tracked for a ray. The ray trace for a ray will be terminated at this number if the ray does not exit the target with fewer reflections. In this case it has been set to 5.
10. Run identifier. This can be used to help verify which GIFT run produced the ray history file.
11. 72 character ascii comment for ray history file header information.
12. This is the name for the ray history file. If the default name is used, GIFT will take the geometry definition or binary file name, remove the extension and use this name with a .ray extension for the ray history file name.
13. Azimuth angle (see Figure 2-6) for the ray trace. This angle along with the elevation angle defines the emanation plane normal.
14. Elevaton angle (see Figure 2-6) for the ray trace. This angle along with the azimuth angle defines the emanation plane normal.
15. This is the number of rays to fire in the horizontal direction.
16. This is the number of rays to fire in the vertical direction.
17. Output from the radar subroutine while it is establishing the emanation plane, the emanation rectangle, and the ray spacing.
18. Verify that the setup is OK and that radar should continue and generate the ray history file.
19. This output indicates that some rays have become "trapped" between two surfaces that have no distance between them. This number of occurances will not harm the ray history.
20. Summary output information from the ray trace. Radar is finished.
21. Prompt for next subroutine selection. User is finished.

```
run [srim.gift]gift                                1
Enter geometry file name: ? dicyii.cg              2

GIFT Program - VAX/VMS FORTRAN 77 Version          3
Program set  1985 February 12
Execution date -  12-APR-85

Input for Main                                     4
s - Print solids        r - Print Regions
i - Print Idents        a - Print Aster array
o - Print ordered id    e - Print ordered reg rpps
t - Overlap tol         l - LOS Tolerance
m - Max errors (25)
Main Option (End=0)? 0

Enter geni                                         5

Title - ft newfangled dicy test object 11-15-84
Target units (in)

Number of solids      9
Number of regions     8

** Diagnostic ** The following regions are null

Total storage for geometry data              248
Total working storage                         35
Total storage in master-aster                275

Tolerance for overlap          tol    =    0.0100
Tolerance for line of sight    tollos =    0.0100

Processed data written on file dicyii.4
Time for input processing    0.00 seconds
Leave geni

Application Subroutines                            6
end     brandx  check   optic   radar
Application Subroutine? radar
```

FIGURE C-2. EXAMPLE GIFT (RADAR) RUN.

```
Enter radar

radar version 07-feb-1985                                  7
modified to keep last ray
print range information
see George Darling about any problems
Bring hardcopy of results


Number of views? 1                                   8
Maximum number of reflections (.lt.255)? 5           9
run identifier? 1                                    10
Enter comment: timing run for gift ray trace         11
Enter ray trace file name: (<retn> = default) dicyii.ray     12
Azimuth (degrees)? 30.          13
Elevation (degrees)? 45.        14
Max horz cells? 128             15
Max vert cells? 128             16

Azimuth  30.0
Elevation  45.0

Target parameters          x          y          z
  Minimum               -14.500    -13.500    -5.100
  Maximum                10.600     15.500     6.000
  Center                 -1.950      1.000     0.450
  Dimensions             25.100     29.000    11.100

View plane
  Horizontal length      37.665
  Vertical length        33.472                                 17
  Depth                  33.472
  Back off distance      17.887
  Center                  1.841      1.159
  Horz Cell size          0.294
  Vert Cell size          0.262
  Horizontal range      -16.991     20.673
  Vertical range        -15.577     17.895

Horz number cells           128
Number vert cells           128
Number of cells           16384
```

FIGURE C-2. EXAMPLE GIFT (RADAR) RUN.
(Continued)

```
Do you r(estart, c(continue or e(nd: ? c          18
     2      3      1         0.000000     0.000000     0.000000
     2      3      1         0.000000     0.000000     0.000000
     2      3      1         0.000000     0.000000     0.000000
     2      3      1         0.000000     0.000000     0.000000     19
     2      3      1         0.000000     0.000000     0.000000
     2      3      1         0.000000     0.000000     0.000000
     2      3      1         0.000000     0.000000     0.000000
     2      3      1         0.000000     0.000000     0.000000
the min and max ranges for 1st surfaces are:          9.512        34.878
The min and max total ranges are:                     9.512        34.878

Time for view     0.00 seconds                                               20

Time for radar     0.00 seconds

Leave radar

Application Subroutines
end     brandx  check   optic   radar                                        21
Application Subroutine? end

End of run
```

FIGURE C-2. EXAMPLE GIFT (RADAR) RUN.
(Concluded)

APPENDIX D
GIFTDUMP WORK SESSION

This Appendix contains a work session for the GIFTDUMP program. The input ray history file for this run was produced by the GIFT run example shown in Figure C-2. The user inputs are underlined and the circled numbers correspond to the numbered notes below. The print file resulting from the run of Figure D-1 is shown in Figure D-2.

Notes for Figure D-1:

1. This is the run command to the VMS operating system.
2. The name of the ray history file to be dumped (from a previous GIFT run.
3. GIFTDUMP outputs a message everytime it reads a physical record from the ray history file. On start up, after reading the first record of the ray history, GIFTDUMP outputs the header information to the terminal.
4. The user wants a histogram of the number of rays with 0,1,2, ... 68 reflections.
5. The user wants GIFTDUMP to only consider rays (for dumping) that have one or more reflections. This does not effect the histogram, only the ray history dump.
6. The user wants specific records dumped (instead of the entire file).
7. The physical record number of the first record to be dumped.
8. GIFTDUMP output while reading up to the tenth record.
9. Prompt for next record to be dumped. A carriage return ‹CR› indicates that no more records are to be dumped. GIFTDUMP continues to read through the ray history file for the histogram information.
10. End message from GIFTDUMP. The information is in a print file named giftdump.lis.

```
run [sria.util]siftdump   1

        enter sift ray trace file name : dicyii.ray   2
read record number  1
**** header information
        -1.00000      -1.00000      -1.95000      1.00000      0.45000
        17.88749      45.00000      30.00000     33.47248     37.66474
       128.00000     128.00000       5.00000      1.00000              3

     timing run for sift ray trace

     run time: 13:38:17              date: 12-APR-85

do you want a reflection histogram? (y/n) : y   4
      minimum number of reflections (i) : 1   5
        do want specific records? (y/n) : y   6
   record number of record to dump (int) : 10   7
read record number  1
read record number  2
read record number  3
read record number  4
read record number  5
read record number  6          8
read record number  7
read record number  8
read record number  9
read record number  10
   record number of record to dump (int) : <CR>  9
read record number  11
read record number  12

        .
        .
        .

read record number  86
read record number  87
read record number  88
read record number  89
read record number  90
read record number  91

normal termination
results are in siftdump.lis     10
FORTRAN STOP
```

FIGURE D-1. EXAMPLE GIFTDUMP RUN.

```
**** ray history dump from file dicyli.ray


**** header information
          -1.00000       -1.00000       -1.75000        1.00000        0.45000
          17.88749       45.00000       30.00000       33.47248       37.66474
         128.00000      128.00000        5.00000        1.00000

      timing run for gift ray trace

      run time: 13:38:17                date: 12-APR-85


*********************** start of ray history dump **********************************


**** record   10 new ray ****

  1 fire        0.00000        2.00000        6.80697        7.24510       15.77917        2.87093
                4.42756       68.00000       52.00000        0.00000        0.00000        0.00000
                0.00000        0.00000        0.00000        0.00000        0.00000

  2 refl        2.00000        1.00000       -6.85819       -0.64448        0.00010        0.00000
                0.00000        1.00000        0.00000       -1.00000        0.00000        0.00000
                0.00000       22.31512        2.00000        2.00000        6.00000

  3 escp        0.00000       -2.00000        0.00000        0.00000        0.00000        0.00000
                0.00000        0.00000       -0.61237       -0.35355        0.70711        0.00000
                0.00000        0.00000        0.00000        0.00000        0.00000

**** record   10 new ray ****

  4 fire        0.00000        2.00000        6.65984        7.49994       15.77917        3.16519
                4.42756       69.00000       52.00000        0.00000        0.00000        0.00000
                0.00000        0.00000        0.00000        0.00000        0.00000

  5 refl        4.00000        1.00000       -1.80917        2.61035        6.00010        0.00000
                0.00000        1.00000        0.00000       -1.00000        0.00000        0.00000
                0.00000       13.82984        5.00000        6.00000        2.00000

  6 escp        0.00000       -2.00000        0.00000        0.00000        0.00000        0.00000
                0.00000        0.00000       -0.61237       -0.35355        0.70711        0.00000
                0.00000        0.00000        0.00000        0.00000        0.00000


***** histogram of reflections per ray

# refl.     count
   0        9696
   1        3954
   2        2247
   3         179
   4          13
   5           4


  **** normal terminaition ****
```

FIGURE D-2. EXAMPLE GIFTDUMP OUTPUT FILE.

APPENDIX E
SHADE WORK SESSION

This Appendix contains a work session for the SHADE program. The ray history used as input is from the GIFT work session shown in Figure C-2. The user inputs are underlined and the circled numbers correspond to the notes given below.

Notes for Figure E-1:

1. This is the run command to the VMS operating system.
2. Prompt for the name of the ray history file. This file must have been created by a previous GIFT run.
3. Header information from the ray history file.
4. Confirmation that the correct ray history file has been opened.
5. Tells the user that the image will be 128 by 128 pixels in size (same as size of the ray grid used in GIFT) and each record in the shaded image fill will be 256 bytes long.
6. The name of the output shaded image file. This will be a file of 16 bit integers.
7. SHADE will use the rav trace direction (azimuth,elevation) as the direction to the illumination source. See OVERLAY example (Figure F-1) for the result of answering "n" to theis prompt.
8. Rectangular components of the unit vector to the illuminator.
9. Integers associated with brigntest, middle, and zero intensity levels in the image.

```
run [srim.shade]shade                                  1
enter input ray trace file : dicyii.ray                2


header record:
run id :         -1
view id:         -1
aim point:      -1.95       1.00       0.45
dist:      17.89                                       3
elevation angle:      45.00
azimuthal angle:    30.000
emanation plane ht and wid:       33.47     37.66
ncols  :         128
nrows  :         128
maxref :           5

correct file? (y/n) :y                                 4


output image will have  128 rows and  128 cols         5
output file will have      256 bytes per record


enter output optical image file :dicyii.mi             6


Do you want default radar direction? (y/n): y          7


normalized light source:
            0.612372            0.353553            0.707107     8


white=      255 gray=      128 black=       0          9
```

FIGURE E-1. EXAMPLE SHADE RUN.

APPENDIX F
OVERLAY WORK SESSION

This Appendix contains an example work session for the OVERLAY program. The ray history used in this work session was produced by the GIFT run shown in Figure C-2. The user inputs are underlined and the circled numbers correspond to the notes below.

Notes for Figure F-1:

1. This is the run command for the VMS operating system.
2. Name of the input ray history file. This file must have been created by a previous GIFT run.
3. Header information from the ray history file.
4. Confirmation that the correct ray history file has been opened.
5. Size of the image in the radar range direction in pixels.
6. Size of the image in the azimuthal radar direction in pixels.
7. Size of an image pixel in distance units (in this case feet). The image will cover 64 feet by 64 feet in the slant range plane.
8. Range offset for positioning target in image (in distance units).
9. Azimuth offset for positioning target in image (in distance units).
10. Size of image in pixels where rows => azimuth dir. and cols => range dir. Also the number of bytes ir an image file record.
11. Name of the output image file.
12. Answer no to using radar direction as direction to illumination source. If the answer was "y", the next two prompts would not be issued.
13. Azimuth direction to illumination source
14. Elevation direction to illumination source.
15. Rectangular components of the illumination unit vector.
16. The integer values assigned to the brightest, middle, and zero intensity levels in the image.

```
run [srim.overlay]overlay                                          1
enter input ray trace file : dicyii.ray                            2

header record:
run id :          -1
view id:          -1
aim point:      -1.95       1.00       0.45
dist:       17.89
elevation angle:      45.00
azimuthal angle:     30.000
emanation plane ht and wid:      33.47     37.66                  3
ncols  :          128
nrows  :          128
maxref :            5

correct file? (y/n) : y                                            4

Enter image range size : 128                                       5
Enter image azimuth size : 128                                     6
Enter pixel spacing : .5                                           7

Enter the range offset : 0.                                        8
Enter the azimuth offset : 0.                                      9

output image will have  128 rows and  128 cols
output file will have      256 bytes per record                   10

enter output optical image file : dicyii.mi                        11

Do you want default radar direction? (y/n): n                      12
Enter azimuth angle 0.                                             13
Enter elevation angle 70.                                          14

normalized light source:
            0.342020            0.000000            0.939693      15

white=     255. gray=     128. black=       0.                     16
```

FIGURE F-1. EXAMPLE OVERLAY RUN.

APPENDIX G
RADSIM INPUT FILES

The three RADSIM input files (other than the ray history file) are the surface type file, the radar file, and the reflection model file. Examples of these files are shown in Figures G-1 to G-3. The files shown in the figures are for the DICYII work session of Appendix I.

The surface type file contains two types of records. The first record in the file contains an integer which is the number of data records that follow. This record is read by an (I5) format. The rest of the file will contain one record for each reflection model assignment to a surface. The number of these records must match the integer in the first record and record contents is the information that assigns a reflection model to a specific surface of the target. The data records are read with a (4I5) format. Note that the integers can be separated by commas in which case they do not need to be in specific columns of the record. This was used in the file shown in Figure G-1. The four items in the data records are:

region number -> The region (as defined in the geometry definition file) that contains the primitive whose surface is to be assigned a reflection model.

body number -> The primitive body that contains the surface (as defined in the geometry definition file).

surface number -> The number of the surface which is to be assigned a surface model.

model number -> The number of the reflection model in the reflection model file that is to be assigned to the surface.

A comparison of the surface type file in Figure G-1 with the geometry definition file of Figure A-3 and the information in Figure A-2 should clraify these items. A drawing of the DICYII target is shown in Figure 4-11. Note that surfaces not appearing in the surface type file will default to reflection model number 1 (a smooth perfect reflector).

The radar file contains the parameters that define the radar system to be modeled. An example radar file is shown in Figure G-2. Each record contains a parameter name and its value. The records are read by the format (A16,F16.6). The records must be in the order shown below and must all be present. Their contents are:

wavlen -> The center wavelength of the radar. For x-band this is 1/10 feet or 3 centimeters. In the example file (Figure G-3) this has been increased by 5% to reduce the effects of undersampling (see Section 4.5.2).

resol -> The distance from the center of the system response to the nearest zero of a sinc function representing the system bandwidth. If a diferent definition of resolution (such as 3db point) or taylor weighting is used, this must be modified

```
17
6,7,5,1          Front of RH wall of trihedral
6,7,2,3          Slanted edge of RH wall of trihedral
6,7,6,3          Back of RH wall of trihedral
6,7,1,3          Back edge of RH wall of trihedral
7,8,5,1          Front of LH wall of trihedral
7,8,2,3          Slanted edge of LH wall of trihedral
7,8,6,3          Back of LH wall of trihedral
7,8,1,3          Back edge of LH wall of trihedral
8,9,1,3          \
8,9,3,3           >   Ground plane edges
8,9,4,3          /
8,9,6,4          Ground plane surface
3,3,1,1          1/4 circular surface (LH side)
4,5,3,1          Inside wall of hollow cylinder
4,5,1,1          Inside floor of hollow cylinder
4,4,3,1          Outside wall of hollow cylinder
4,4,2,1          Top (ring) of hollow cylinder
```

FIGURE G-1. ANNOTATED SURFACE TYPE FILE. (Note that the annotation does not exist in the data file.)

WAVLEN	.105
RESOL	5.0
XVPOL	1.
XHPOL	0.
RVPOL	1.
RHPOL	0.
SPACEN	2.5
NCIS	0.
SLL	-30.
TPRSIZ	0.
ATAYLOR	1.
CONV. CELLS	11.
QUANT. FACT.	10.
FRACT BW	0.392

FIGURE G-2. EXAMPLE RADAR FILE.

5

	MU	EPS	RHO	ROUGH	ABSORB	POLSCA	PHASCA
1	0.,	0.,	0.,	0.,	0.,	0.,	0.
2	0.,	0.,	0.,	0.,	.5,	0.,	0.
3	0.,	0.,	0.,	0.,	1.0,	0.,	0.
4	0.,	15.0,	0.,	4.0,	0.,	0.,	0.
5	5.0,	12.0,	6.0,	3.0,	0.,	.5,	1.0

FIGURE G-3. EXAMPLE REFLECTION MODEL FILE.

appropriatly.

xvpol -› The vertical polarization of the transmitter. This and the value for xhpol are used to compute a transmitter unit vector polarization direction in the emanation plane.

xhpol -› The horizontal polarization of the transmitter. This and the value for xvpol are used to compute a transmitter unit vector polarization direction in the emanation plane.

rvpol -› The vertical polarization of the receiver. This and the value for rhpol are used ot compute a receiver unit vector polarization direction in the emanation plane.

rhpol -› The horizontal polarization of the receiver. This and the value for rvpol are used to compute a receiver unit vector polarization direction in the emanation plane.

spacen -› The size of a pixel in the slant range plane. In the example file, a pixel will cover 2.5 feet by 2.5 feet in the slant range plane. The image will be samples twice per resolution (5/2.5) and the image will cover 320 feet by 320 feet in the slant plane.

nois -› Average noise power in the image in db. If this is non-zero, then the complex image will be initialized with random Gaussian noise and the target returns will be coherently summed with this noise. If this number is zero (as it is in the example file), the complex image is initialized to zeros.

sll -› This is the level (relative to the main lobe) of the first sidelobe in the syystem response. This is used to specifiy the sidelobe level when a Taylor weighted system response is used. It is ignored if a sinc sstem response is used.

iprsiz -› Not used.

ntaylor -› The number of terms to use in the expansion for a Taylor weighted system response. A value of 1 results in a sinc system response while a value › 1 gives a Taylor weighted response with this number of terms in the expansion.

conv. cells -› The number of pixels that will be spanned by the system response. In the example file, the system response will span 5 pixels on either side of the center pixel.

quant. fact. -› Quantization factor for the system response. When the system response is digitized, it will be sampled this many times per pixel. In the example file, the system response will be sampled 10 times per pixel resulting in 55 samples representing one side of the symmetric system response.

fract bw -› This is only used for an experimental function and is ignored.

The reflection model file contains the information that

defines the reflection models. These are the models assigned to the surfaces in the surface type file. The file currently used is shown in Figure G-3. The file contains three types of records. The first record in the file gives the number of models that are defined in the file. This record is read with a (I5) format. The second record contains an ASCII string used as a label and is read with an (A80) format. The next records define the reflection models. there is one record for each reflection model and the number of records must match the integer in the first record. Each reflection model definition record contains information specific to that model and is read as (a2,10f15.0). The first two characters are the model number used in the surface type file. The other information is model dependent and is given in Appendix H.

APPENDIX H
RADSIM REFLECTION MODELS

This appendix presents the information used to define the various reflection models. The information is in the reflection model file which was discussed in Appendix I. The current reflection model file is shown in Figure G-3 and contains five models. The first three models are for smooth reflectors differing only in their values for "ABSORB". All other values are zero. The first model is a perfect reflector (such as a smooth perfectly conducting metal surface), the second model is for a smooth reflector which reduces the amplitude of the incident radiation by 1/2, and the third model is for a surface which totally absorbs any incident radiation.

The last two models (4 and 5) represent rough surface models and are used for ground clutter. The two rough surface models are: a statistical model which produces random speckle (or clutter) in the output image with the same statistics as speckled surfaces in actual SAR imagery, and a fractal model which is a first- order approximation of the small (ie. on the order of a wavelength) structure of the scattering surface.

The statistical model models the complex returns from a rough surface as independent, identically distributed, zero-mean, gaussian random numbers. For this model, whenever a ray intersects a rough surface, random zero-mean gaussian numbers are used to determine the real and imaginary parts of the return. The amplitude of the complex return the surface would generate if it was a smooth reflector is added to the amplitude of the randomly generated return. Control over the degree of roughness is obtained by selecting the standard deviation of the gaussian distribution used to select the random numbers. The following assignments for the data in the reflection model file have been made:

ROUGH -> must be set to 4 to select this model.

EPS -> Standard deviation of the random numbers.

The values used for the standard deviation is EPS = 15.

The fractal model is implimented as follows (see Figure H-1). If a ray that is being traced by RADSIM reflects from a fractal surface, then at the point of intersection (xi,yi,zi) a plane is created which is perpendicular to the surface normal at that point (N). A grid is superimposed on the plane and at each of the grid points a height is assigned based on a fractal model. The complex return to the radar for the reflection is calculated as if the surface were a smooth reflector using the user-defines absorbtion coefficient (ABSORB). Then, at each grid point a complex return is calculated which has the same amplitude as the original return but whose phase is modified based on the difference in the distance from the ray's starting point to the intersection point (D1) and the distance from the ray's starting point to the current grid point (Dg). The complex

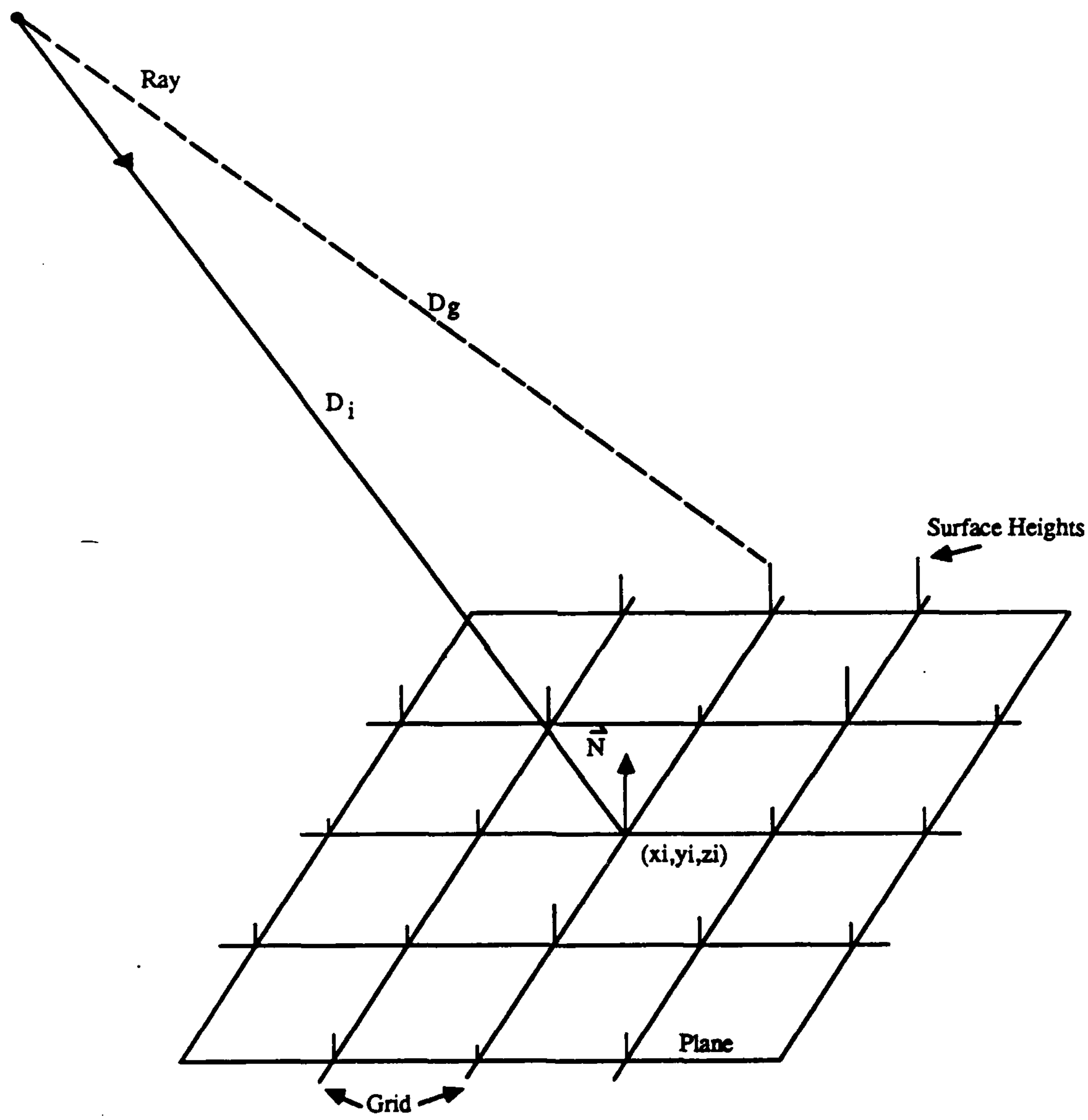


FIGURE H-1. FRACTAL SURFACE FOR SPECKLE

returns from each of the grid points are added coherently to produce the actual return.

To specify this model the following five parameters are needed: the lengths of the sides of the plane; the number of grid points to place on the plane; and three parameters to specify the fractal model. The fractal model parameters are the correlation between adjacent fractal height differences, the variance of the fractal heights, and the maximum fractal height. It has been found that changes in the maximum fractal height have the most effect on the clutter, while changes in the other parameters had little effect (assuming that those changes were not large enough to elliminate the random nature of the fractal surface all together). This can be explained intuitively as follows. First, in this implementation we are using the fractal model to introduce randomness into the complex returns, not to actually model the surface. Thus it is the overall randomness of the entire fractal surface which produces a single complex return and fluctuations in the local randomness will have a small affect on the overall randomness. Second, it is the phase differences between returns from the grid points which produce the speckle and if the maximum fractal height (which scales the whole fractal surface) is such that the phase differences produced are much less than a wavelength, no speckle will be seen. On the other hand, if they are much greater than a wavelength speckle will be visible and if they are much greater than a wavelength the returns are distributed uniformly in [-pi,pi] which closely models speckle in actual SAR images. It has been found that good speckle can be produced with the following parameters:

ROUGH -› This is set to 3. to indicate a fractal model.

MU -› The number of grid points along a side of the plane (currently set to 5 which implies 25 grid points).

EPS -› The length of a side of the plane (currently set to 12 wavelengths)

RHO -› The maximum fractal height (currently set to 6 wavelengths).

ABSORB -› Same as for smooth reflectors.

POLSCA -› The correlation parameter (currently set to .5).

PHASCA -› The variance parameter (currently set to 1.).

Different values of the maximum fractal height are used to produce different random surfaces.

APPENDIX I
RADSIM WORK SESSION

This appendix contains a RADSIM work session. The input ray history file is from the GIFT work session shown in Figure C-2. The other input files are covered in Appendix G and Appendix H. The user inputs are underlined and the circled numbers correspond to the notes below.

Notes for Figure I-1:

1. This is the run command for the VMS operating system.
2. These are the single and double precision tolerances calculated by RADSIM at program startup.
3. The name of the input ray history file produced by a previous GIFT run.
4. Ray history file header information.
5. Confirmation that the correct ray history file has been opened.
6. Emanation plane unit vectors in the vertical, horizontal and normal directions.
7. Name of the surface type file for the target. The target used for this run is the DICYII target.
8. This is a list of the surface type file contents.
9. Confirmation that this is the correct surface type file.
10. The name of the radar definition file. A carriage return would cause RADSIM to use the default radar file name "radar.dat". A ‹CR› could have been used since the name given is the same as the default name.
11. Listing of the contents of the radar definition file.
12. Confirmation that the correct radar file has been opened.
13. RADSIM will output a complex image file. An answer of "r" would result in RADSIM outputing a scaled real image file.
14. The range and azimuth size of the image in pixels. Note that the radar file specifies SPACEN as .5. Thus the image will cover 64 feet by 64 feet in the slant range plane.
15. Offsets for positioning the target in the image. A carriage return ‹CR› causes them to be zero which centers the overall coordinate system origin in the image.

16 Target positioning information computed from ray history header information and the offsets.

17. Name of the output complex image file.
18. RADSIM will create a new image file. An answer of "a" would cause RADSIM to attempt to attach an existing complex image file and add to the image using the current ray trace. This cannot be done with a scaled real image file.
19. Name of the reflection model file. A carriage return ‹CR› causes the default file name "surface.dat" to be

used.

20. Listing of the contents of the reflection model file.
21. Confirmation that this is the correct reflection model file.
22. The user does not wish to define zones of interest. If the answer had been "y", then RADSIM would have prompted for the zones of interest as shown in Figure 3-22.
23. RADSIM end message giving the number of records read from the ray history file, the total complex return and radar cross section, and the total number of returns found for the run. If a scaled real image fill was output, RADSIM would print out the scale factor as a part of its ending message.

```
run [srim.radsim]radsim                                                        1
single precision eps =              0.22204460E-15            = 2(- 52)
double precision eps =       0.516987882845642D-25            = 2(- 84)         2

enter ray history file name:dicyii.ray       3

ay history file header record:
                 -1 run id             -1 view id
             -1.950                  1.000               0.450          aim point
             17.887 dist            45.000 el           30.000 az                  4
             33.472 ht              37.665 wd
                128 nrow               128 ncol
                  5 maxref               1 shot

correct ray history file? [y/n]y     5

vertical emanation vector =       0.612372E+00   0.353553E+00  -0.707107E+00
horizontal emanation vector =     0.500000E+00  -0.866025E+00   0.000000E+00   6
normal vector =                  -0.612372E+00  -0.353553E+00  -0.707107E+00

Enter surface type file name :       dicyii.sur        7

Surface parameters are:
numsur =     17
index     region     body type     surface type     model number
    1          6             7                5                1
    2          6             7                2                3
    3          6             7                6                3
    4          6             7                1                3
    5          7             8                5                1
    6          7             8                2                3
    7          7             8                6                3
    8          7             8                1                3              8
    9          8             9                1                3
   10          8             9                3                3
   11          8             9                4                3
   12          8             9                6                4
   13          3             3                1                1
   14          4             5                3                1
   15          4             5                1                1
   16          4             4                3                1
   17          4             4                2                1

Correct surface file type? [y/n] : y     9
```

FIGURE I-1. EXAMPLE RADSIM RUN.

```
Enter radar file name
  [default is 'radar.dat'] : radar.dat    10

 WAVLEN                  0.105000
 RESOL                   5.000000
 XVPOL                   1.000000
 XHPOL                   0.000000
 RVPOL                   1.000000
 RHPOL                   0.000000
 SPACEN                  2.500000
 NOIS                    0.000000                 11
 SLL                   -30.000000
 IPRSIZ                  1.000000
 NTAYLOR                 1.000000
 CONV. CELLS            11.000000
 QUANT. FACT.           10.000000
 FRACT BW                0.392000
11 pixels will be used for range convolution
correct file? [y/n] : y   12
scaled magnitude or complex image ? [m/c]  c   13
enter image size:  [2.5: range,azimuth]128,128 14
enter range and azimuth offsets: [2e10.5] <CR> 15

  scale=     0.20000E+01
  azthr=    -0.16000E+03               16
halfrng=     0.16000E+03
 center=     0.17987E+02

enter output image file name: dicsii.ci         17
create new file or attach existent? (c/a):c     18

Enter reflection model file name
  [default is 'surface.dat'] :   <CR>     19

Reflection model file contains   5 models.
Modtyp array references  4.

   MU        EPS      RHO      ROUGH    ABSORB   POLSCA  PHASCA

1   0.0000   0.0000   0.0000   0.0000   0.0000   0.0000   0.0000        20
2   0.0000   0.0000   0.0000   0.0000   0.5000   0.0000   0.0000
3   0.0000   0.0000   0.0000   0.0000   1.0000   0.0000   0.0000
4   0.0000  15.0000   0.0000   1.0000   0.0000   0.0000   0.0000
5   5.0000  12.0000   0.0000   3.0000   0.0000   0.5000   1.0000
correct file? [y/n]:y    21
define zones of interest in image? [y/n]n    22
done number records =          91
 totali = -0.150E+04  totalq =  0.170E+03  rcs =  0.227E+07  ptncnt =    9738    23
FORTRAN STOP
```

FIGURE I-1. EXAMPLE RADSIM RUN.
(Concluded)

APPENDIX J
DETECT WORK SESSION

This Appendix contains a work session for the DETECT program. The input complex image was produced by the RADSIM run example shown in Figure I-1. The user inputs are underlined and the circled numbers correspond to the numbered notes below.

Notes for Figure J-1:

1. This is the run command to the VMS operating system.
2. This selects the "set scale" option in DETECT. This should be the first option selected when DETECT is run.
3. This selects "autoscaling" for the scale factor. This scale factor will be multiplied by the magnitude of the complex numbers before the magnitude is converted to an integer in the range 0 to 255.
4. This selects the "set input file" option.
5. The name of the file containing the complex image to be detected.
6. This selects the "detect image" option.
7. This is the name of the output file that the detected image will be written to. DETECT now detects the image and writes the magnitude file.
8. This ends the DETECT program and returns the user to the VMS operating system.

```
run [srim.detect]detect                                   1

***SRIM detection program***


Complex image file :
Complex image maximum magnitude :     0.000000E+00
User specified scaling factor :       0.000000E+00

Commands :
 [F] - set input file
 [S] - set scale
 [D] - detect image
 [E] - exit program

Enter command : s                          2
(A)utoscale or (M)anual? : a               3

Complex image file :
Complex image maximum magnitude :     0.000000E+00
Autoscaling factor :                  0.255000E+03

Commands :
 [F] - set input file
 [S] - set scale
 [D] - detect image
 [E] - exit program

Enter command : f                                4
Enter input filename : dicyii.ci                 5

Complex image file :                dicyii.ci
Complex image maximum magnitude :     0.845600E+03
Autoscaling factor :                  0.301561E+00

Commands :
 [F] - set input file
 [S] - set scale
 [D] - detect image
 [E] - exit program

Enter command : d                                  6
Enter output filename : dicyii.mi                  7

Complex image file :                dicyii.ci
Complex image maximum magnitude :     0.845600E+03
Autoscaling factor :                  0.301561E+00

Commands :
 [F] - set input file
 [S] - set scale
 [D] - detect image
 [E] - exit program

Enter command : e
$                                                  8
```

FIGURE J-1. EXAMPLE DETECT WORK SESSION.